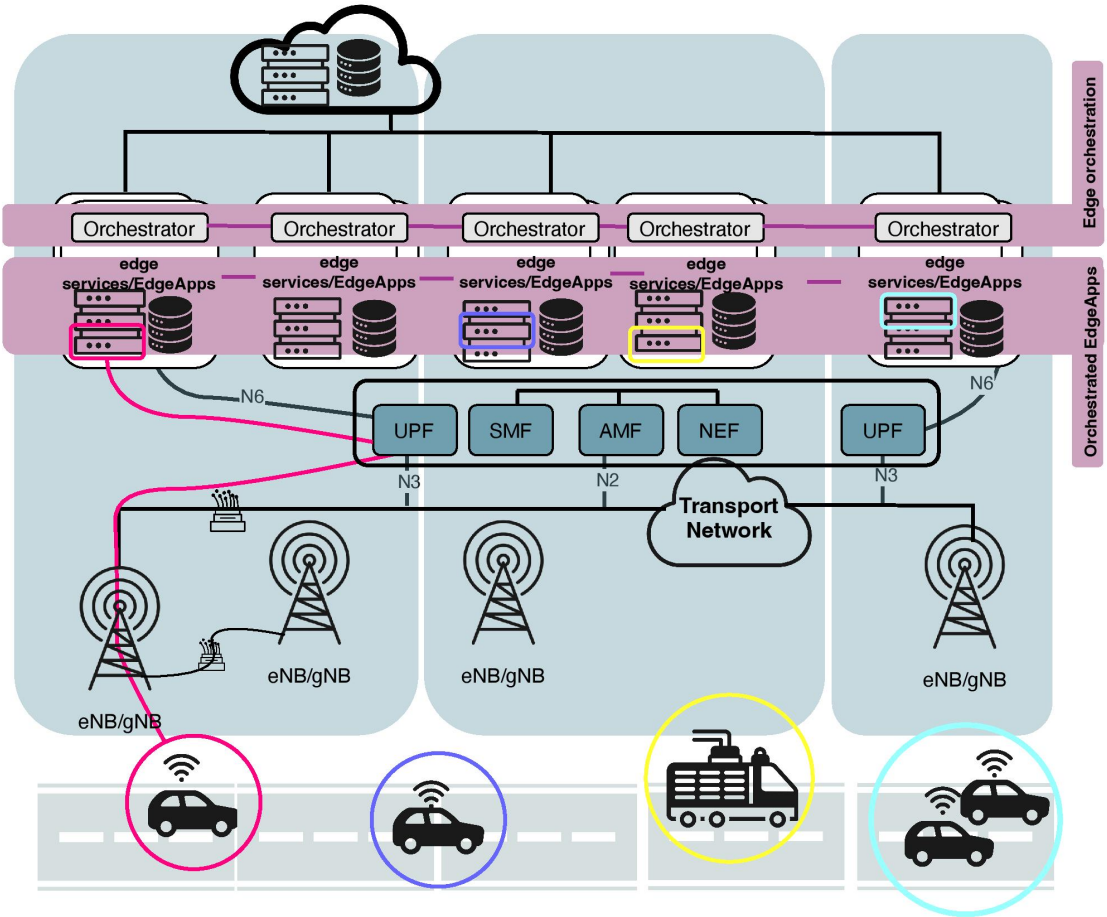
Orchestrating 5G Edges: Collaborative and Distributed Management and Orchestration of Open Programmable and Virtualized Edge Networks

Nina Slamnik-Kriještorac



Supervisors **Prof. dr. Johann M. Marquez-Barja — Prof. dr. Steven Latré**

Thesis submitted in fulfilment of the requirements for the degree of Doctor in Applied Engineering
Faculty of Applied Engineering — Antwerp, 2022

University of Antwerp

Faculty of Applied Engineering
Doctor in Applied Engineering

Orchestrating 5G Edges: Collaborative and Distributed Management and Orchestration of Open Programmable and Virtualized Edge Networks

Thesis submitted in fulfilment of the requirements for the degree of
Doctor in Applied Engineering
at University of Antwerp

Nina Slamnik-Kriještorac

Supervisors
Prof. dr. Johann M. Marquez-Barja
Prof. dr. Steven Latré

Antwerp, 2022

Jury

Chairman
Prof. dr. Jeroen Famaey, University of Antwerp, Belgium

Supervisors
Prof. dr. Johann M. Marquez-Barja, University of Antwerp, Belgium
Prof. dr. Steven Latré, University of Antwerp, Belgium

Members
Dr. Rafael Berkvens, University of Antwerp, Belgium
Prof. dr. Michael Peeters, University of Antwerp, IMEC, Belgium
Dr. Faqir Zarrar Yousaf, NEC Laboratories Europe, Germany
Prof. dr. Claudio E. Palazzi, University of Padua, Italy

Contact
Nina Slamnik-Kriještorac
University of Antwerp
Faculty of Applied Engineering - Electronics-ICT
IDLab research group
Sint-Pietersvliet 7, 2020 Antwerp, Belgium
M: nina.slamnik-krijestorac@uantwerpen.be

ISBN 978-90-57287-70-1
Wettelijk depot D/2022/12.293/40

Abstract

The digital era we live in brings numerous opportunities for societies by improving safety, transportation, as well as health and well-being. To achieve such benefits, it is required that all technological elements such as sensors, phones, vehicles, and facilities, are connected and able to exchange information via the network. However, such ubiquitous connectivity inevitably involves a tremendous increase in network traffic and the need for ultra-low latencies, which are well-recognized challenges in both research and industry. To accommodate new digital systems and services, and to respond to innovation aspirations coming from the digital society, networks need to be significantly more flexible and adjustable. Such flexibility can be achieved through network softwarization, which ensures that network infrastructure is fully programmable and adjustable. The two main pillars of network programmability are Network Function Virtualization (NFV) and Software Defined Networking (SDN), where the former virtualizes network resources and enables the dynamic creation of Virtualized Network Functions (VNFs), and the latter programs the way those VNFs are connected to each other over the network.

The 5G ecosystems, consisting of 5G New Radio, edge, transport, and core network, are robust software-based systems that rely on both SDN and NFV to enable network programmability. By applying the concept of virtualization (e.g., in the core and edge networks, and partially on the radio side), more and more resources are virtualized, resulting in an enormous heterogeneity across the network ecosystems. Thus, one of the significant challenges lies in agile management and orchestration of those resources that are diverse and distributed across the overall 5G ecosystem. The MANagement and Orchestration (MANO) role is to identify the needs of traffic and services running on the virtualized network infrastructure and dynamically respond to those needs by performing service reconfigurations, such as scaling, migration, and service teardown, which are known as MANO operations. Throughout this PhD research, we have been focused on investigating service and resource management and orchestration in Multi-Access Edge Computing (MEC)-enhanced 5G networks to enable openness and programmability of 5G and beyond ecosystems. The main objective of this research is to leverage 5G, MEC, and Artificial Intelligence (AI)/Machine Learning (ML), to perform efficient and automated MANO operations across different technological/administrative edge domains, and to achieve the low-latency-aware VNF placement and seamless migration of programmable Network Services (NSs).

We started this research by investigating the 5G network resources and service programmability from a broader perspective, entailing both wireless and optical domains, altogether with the edge and cloud, which are all indispensable parts of the 5G network ecosystems. In particular, we focused on resource sharing, as a paradigm that shifts the exclusive ownership of network resources toward mutual resource use that enables service performance improvements and cost savings at the same time. This part of the research work focused

on: i) identifying the types of resources as well as the techniques that need to be applied to enable sharing, and ii) studying the challenges that imply from the overall resource sharing. Having learned that NFV and SDN are enabling utilization and management of resources in a flexible and programmable manner, we moved from a resource sharing perspective to a comprehensive and immensely challenging resource and service management and orchestration of distributed 5G edge networks. Thus, we studied the closed-loop life-cycle management of 5G services, which consists of three main intertwined phases, i.e., orchestration, control, and monitoring. These three phases are backed by SDN and NFV that bring more flexibility, and programmability to wired and wireless communication networks, while enabling higher resource utilization, and lower costs. As a part of this research, we conducted extensive experimentation using a real testbed setup (Virtual Wall testbed, Ghent, Belgium), thereby analyzing the performance of various existing NFV MANO solutions, and studying their suitability and readiness for orchestrating vehicular services in distributed edge environments, later studied on top of the Smart Highway testbed (Antwerp, Belgium).

Being one of the most challenging consumers of the 5G ecosystems, the automotive industry and its connected vehicles require more and more support from the network and virtualized infrastructure to achieve connectivity with low-latency (1-10ms), high-reliability (99,999%), enhanced throughput (up to 20Gbps), and efficient resource usage. Due to such challenging nature of vehicular communications, our research on service management and orchestration steered toward this particular and challenging type of vertical service. The deployment of service instances at distributed resources of cellular network infrastructure edges enables localized low-latency access to these services from moving vehicles but comes along with challenges, such as the need for fast reconfiguration of the distributed deployment according to mobility patterns and associated services, and resource demand. To this end, we investigated and proposed a solution for the collaborative orchestration of services for Connected, Cooperative and Automated Mobility (CCAM) within such a 5G ecosystem, with the key objective to ensure service continuity for a highly dynamic automotive scenario, through performing associated management and orchestration of these services in distributed edge clouds. The performance evaluation of the orchestration systems has been conducted in various distributed proof-of-concept setups where we utilized the Virtual Wall and CityLab testbeds located in Ghent and Antwerp, respectively.

To showcase the operations of such collaborative orchestration in distributed edge environments, and their impact on the service performance, we have been extensively working on a use case that aims to enhance mission-critical services by provisioning VNFs at the network edge, i.e., the Back-Situation Awareness (BSA) that supports Emergency Vehicles (EmVs) by increasing awareness about them on the roads. The performance evaluation of this 5G V2X use case, as well as its orchestration, has been performed utilizing the Smart Highway testbed, where the Roadside Units (RSUs) installed along the E313 highway (Antwerp, Belgium) served as distributed edge computing nodes. In addition, applying the cloud-native principles and programmability of service function chains to the design and development of use cases in 5G ecosystems, we have not only worked on the aforementioned BSA application service, but also on all-encompassing Edge Network Applications (EdgeApps). Such EdgeApps build any complex 5G vertical service (e.g., automotive, and transport & logistics), by abstracting the underlying 5G network complexity, and thus bridging the knowledge gap between vertical stakeholders, network experts, and application developers.

In the final phase of this PhD research, we have been focused on bringing automation and intelligence to MANO operations, which are inevitable given the complexity of MANO systems of the 5G and beyond services. Such complexity demands innovative approaches to remove limitations of existing techniques, as these techniques might cause a large delay in MANO operations, and thus, negatively impact the service performance. If we consider the traditional techniques, such as the human-in-the-loop approach that is slow and prone to errors, where taking actions might take too long, and the closed-loop control using rule-based algorithms that is difficult to design (an abundant number of parameters need to be configured), it becomes hardly feasible to efficiently orchestrate complex and dynamic vehicular environments. Thus, applying AI/ML in combination with NFV and SDN seems a promising solution for enabling automation and intelligence that will optimize MANO operations. To this end, we extensively studied the gaps in current NFV MANO solutions for efficient orchestration of 5G vehicular edge services, and based on such gaps, proposed an AI/ML-based closed-loop control framework for NFV MANO system, thereby identifying the specific AI/ML techniques that can alleviate the identified gaps and studying the implications resulting from applying certain AI/ML techniques as Network Intelligence Functions (NIFs). Finally, by applying some of the studied and identified techniques, we have built a proof-of-concept using real-life testbeds, both Smart Highway and Virtual Wall, to prove and validate the benefits of AI-enhanced algorithms for automated and intelligent management and orchestration. Enabling automatic and service-agnostic management and orchestration requires the creation of novel architecture of loosely coupled management and orchestration elements, with open and programmable interfaces that will enable the communication between those elements and NIFs running inside the distributed network segments. Thus, following the work executed by standardization bodies, such as European Telecommunications Standardization Institute (ETSI), in the two working groups: Zero-touch network and Service Management (ZSM), and Experiential Networked Intelligence (ENI), this research contributes to accelerating the automatic execution of MANO operations by injecting the intelligence into various network segments (e.g., edge services/EdgeApps, orchestrators, platform managers, radio, and core network), which is even going beyond the scope of current standardization activities in these two working groups. This research opens up the potential of extending the standardization activities and thus contributing to standardization bodies.

One of the main components of this research is its applicability to the next-generation real-world networks and systems, as it has been driven by an active engagement with industry, which resulted in a strong collaboration with several important partners from 5G, automotive, and transport & logistics industries. Moreover, the thesis contributed to the following European projects: H2020 5G-CARMEN, H2020 DAEMON, H2020 VITAL-5G, H2020 5G-BLUEPRINT, and H2020 FED4FIRE+, resulting in five journal papers (published and submitted), 12 publications in conference proceedings (full, work-in-progress, and demo papers; published and submitted), and a book chapter, as a first author, as well as three journals and six conference papers, as a co-author (published and submitted).

Samenvatting

Het digitale tijdperk waarin wij leven, biedt talrijke mogelijkheden voor samenlevingen door verbetering van de veiligheid, het vervoer, de gezondheid en het welzijn. Om dergelijke voordelen te bereiken is het nodig dat alle technologische elementen, zoals sensoren, telefoons, voertuigen en faciliteiten, met elkaar verbonden zijn en informatie kunnen uitwisselen via het netwerk. Een dergelijke alomtegenwoordige connectiviteit gaat echter onvermijdelijk gepaard met een enorme toename van het netwerkverkeer en de noodzaak van ultra-lage latenties, wat zowel in het onderzoek als in de industrie algemeen erkende uitdagingen zijn. Om nieuwe digitale systemen en diensten aan te kunnen en te kunnen inspelen op de innovatieaspiraties van de digitale samenleving, moeten netwerken aanzienlijk flexibeler en aanpasbaarder zijn. Een dergelijke flexibiliteit kan worden bereikt door netwerksoftwarisering, die ervoor zorgt dat de netwerkinfrastructuur volledig programmeerbaar en aanpasbaar is. De twee belangrijkste pijlers van netwerkprogrammeerbaarheid zijn Network Function Virtualization (NFV) en Software Defined Networking (SDN), waarbij de eerste de netwerkbronnen virtualiseert en de dynamische creatie van gevirtualiseerde netwerkfuncties of Virtual Network Functions (VNF's) mogelijk maakt, en de tweede de manier programmeert waarop die VNF's via het netwerk met elkaar worden verbonden.

De 5G-ecosystemen, bestaande uit 5G New Radio, edge, transport en core netwerk, zijn robuuste softwaregebaseerde systemen die steunen op zowel SDN als NFV om netwerkprogrammeerbaarheid mogelijk te maken. Door de toepassing van het virtualisatieconcept (bv. in de kern- en randnetwerken, en gedeeltelijk aan de radiokant) worden steeds meer middelen gevirtualiseerd, wat resulteert in een enorme heterogeniteit in de netwerkecosystemen. Een van de belangrijke uitdagingen ligt dus in een soepel beheer en orkestratie van die middelen die divers en verspreid zijn over het gehele 5G-ecosysteem. De rol van MANO (Management and Orchestration) is het identificeren van de behoeften van het verkeer en de diensten die op de gevirtualiseerde netwerkinfrastructuur draaien, en het dynamisch inspelen op die behoeften door dienstreconfiguraties uit te voeren, zoals schaling, migratie en dienstafbraak, die bekend staan als MANO-operaties. Gedurende dit doctoraatsonderzoek hebben wij ons gericht op het onderzoeken van diensten- en middelenbeheer en orkestratie in Multi-Access Edge Computing (MEC)-uitgebreide 5G-netwerken om openheid en programmeerbaarheid van 5G en verdere ecosystemen mogelijk te maken. De belangrijkste doelstelling van dit onderzoek is het gebruik van 5G, MEC, en Artificial Intelligence (AI)/Machine Learning (ML), om efficiënte en geautomatiseerde MANO-operaties uit te voeren over verschillende technologische/administratieve edge domeinen, en om de low-latency-aware VNF-plaatsing en naadloze migratie van programmeerbare Network Services (NSs) te bereiken.

Wij begonnen dit onderzoek door de 5G-netwerkbronnen en de programmeerbaarheid van diensten vanuit een breder perspectief te onderzoeken, waarbij zowel draadloze als optische domeinen worden betrokken, samen met de edge en cloud, die alle onmisbare onderdelen

zijn van de 5G-netwerkecosystemen. In het bijzonder richtten wij ons op het delen van hulpbronnen, als een paradigma dat het exclusieve eigendom van netwerkhulpbronnen verschuift naar wederzijds gebruik van hulpbronnen dat tegelijkertijd prestatieverbeteringen en kostenbesparingen mogelijk maakt. Dit deel van het onderzoekswerk was gericht op: i) het identificeren van de soorten middelen en de technieken die moeten worden toegepast om gedeeld gebruik mogelijk te maken, en ii) het bestuderen van de uitdagingen die voortvloeien uit het delen van middelen. Nu we hebben geleerd dat NFV en SDN het gebruik en beheer van middelen op een flexibele en programmeerbare manier mogelijk maken, zijn we van het perspectief van het delen van middelen overgestapt op een uitgebreid en immens uitdagend beheer van middelen en diensten en orkestratie van gedistribueerde 5G-randnetwerken. Zo bestudeerden wij het levenscyclusbeheer van 5G-diensten, dat bestaat uit drie belangrijke, met elkaar verweven fasen, namelijk orkestratie, controle en toezicht. Deze drie fasen worden ondersteund door SDN en NFV, die meer flexibiliteit en programmeerbaarheid brengen in bedrade en draadloze communicatienetwerken, terwijl ze een hoger gebruik van middelen en lagere kosten mogelijk maken. Als onderdeel van dit onderzoek hebben we uitgebreide experimenten uitgevoerd met een echte testbed opstelling (Virtual Wall testbed, Gent, België), waarbij we de prestaties van verschillende bestaande NFV MANO oplossingen hebben geanalyseerd en hun geschiktheid en gereedheid voor het orkestreren van voertuigdiensten in gedistribueerde randomgevingen hebben bestudeerd, later bestudeerd bovenop de Smart Highway testbed (Antwerpen, België).

Als een van de meest uitdagende gebruikers van de 5G-ecosystemen vereisen de autoindustrie en haar verbonden voertuigen steeds meer ondersteuning van het netwerk en de gevirtualiseerde infrastructuur om connectiviteit met lage latentie (1-10 ms), hoge betrouwbaarheid (99,999%), verbeterde doorvoer (tot 20 Gbps) en efficiënt gebruik van middelen te bereiken. Vanwege deze uitdagende aard van voertuigcommunicatie is ons onderzoek naar dienstenbeheer en -orkestratie gericht op dit specifieke en uitdagende type verticale dienst. De inzet van diensteninstanties op gedistribueerde bronnen van cellulaire netwerkinfrastructuurranden maakt gelokaliseerde toegang met lage latentie tot deze diensten vanuit rijdende voertuigen mogelijk, maar gaat gepaard met uitdagingen, zoals de noodzaak van snelle herconfiguratie van de gedistribueerde inzet volgens mobiliteitspatronen en bijbehorende diensten, en de vraag naar middelen. Daartoe hebben wij een oplossing onderzocht en voorgesteld voor de collaboratieve orkestratie van diensten voor Connected, Cooperative and Automated Mobility (CCAM) binnen een dergelijk 5G-ecosysteem, met als hoofddoel de continuïteit van de dienstverlening te garanderen voor een zeer dynamisch automobielscenario, door het bijbehorende beheer en de orkestratie van deze diensten in gedistribueerde edge clouds uit te voeren. De prestatie-evaluatie van de orkestratiesystemen werd uitgevoerd in verschillende gedistribueerde proof-of-concept-opstellingen waarbij we gebruik maakten van de Virtual Wall en CityLab testbeds in respectievelijk Gent en Antwerpen.

Om de werking van dergelijke collaboratieve orkestratie in gedistribueerde randomgevingen en hun impact op de dienstprestaties te demonstreren, hebben wij uitgebreid gewerkt aan een use case die tot doel heeft missiekritische diensten te verbeteren door VNF's aan te bieden aan de rand van het netwerk, d.w.z. de Back-Situation Awareness (BSA) die Emergency Vehicles (EmV's) ondersteunt door hun bewustzijn op de wegen te vergroten. De prestatie-evaluatie van deze 5G V2X use case, evenals de orkestratie ervan, is uitgevoerd met behulp van de Smart Highway testbed, waar de Roadside Units (RSU's) geïnstalleerd langs de E313 snelweg (Antwerpen, België) dienden als gedistribueerde edge computing

nodes. Bovendien hebben wij, door de cloud-native principes en de programmeerbaarheid van dienstfunctieketens toe te passen op het ontwerp en de ontwikkeling van use cases in 5G-ecosystemen, niet alleen gewerkt aan de bovengenoemde BSA-toepassingsdienst, maar ook aan allesomvattende Edge Network Applications (EdgeApps). Dergelijke EdgeApps bouwen elke complexe verticale 5G-dienst (bv. automobielsector, en transport en logistiek), door de onderliggende 5G-netwerkcomplexiteit te abstraheren en zo de kenniskloof tussen verticale belanghebbenden, netwerkexperts en applicatieontwikkelaars te overbruggen.

In de laatste fase van dit doctoraatsonderzoek hebben we ons gericht op het automatiseren en intelligent maken van MANO-operaties, wat onvermijdelijk is gezien de complexiteit van MANO-systemen van de 5G- en volgende diensten. Een dergelijke complexiteit vraagt om innovatieve benaderingen om de beperkingen van bestaande technieken weg te nemen, aangezien deze technieken een grote vertraging kunnen veroorzaken in MANO-operaties, en dus de prestaties van de dienst negatief kunnen beïnvloeden. Als we kijken naar de traditionele technieken, zoals de human-in-the-loop aanpak die traag en foutgevoelig is, waarbij het nemen van maatregelen te lang kan duren, en de closed-loop controle met behulp van regelgebaseerde algoritmen die moeilijk te ontwerpen is (een overvloedig aantal parameters moet worden geconfigureerd), wordt het nauwelijks haalbaar om complexe en dynamische voertuigomgevingen efficiënt te orkestreren. Het toepassen van AI/ML in combinatie met NFV en SDN lijkt dus een veelbelovende oplossing om automatisering en intelligentie mogelijk te maken die MANO-operaties zullen optimaliseren. Daartoe hebben we de hiaten in de huidige NFV MANO-oplossingen voor efficiënte orkestratie van 5G-randdiensten voor voertuigen uitgebreid bestudeerd, en op basis van die hiaten een AI/ML-gebaseerd gesloten regelkader voor NFV MANO-systeem voorgesteld, waarbij we de specifieke AI/ML-technieken hebben geïdentificeerd die de geïdentificeerde hiaten kunnen opvullen en de implicaties hebben bestudeerd van de toepassing van bepaalde AI/ML-technieken als Network Intelligence Functions (NIF's). Ten slotte hebben wij, door enkele van de bestudeerde en geïdentificeerde technieken toe te passen, een proof-of-concept gebouwd met gebruikmaking van real-life testbeds, zowel Smart Highway als Virtual Wall, om de voordelen van AI-versterkte algoritmen voor geautomatiseerd en intelligent beheer en orkestratie te bewijzen en te valideren. Om automatisch en dienstagnostisch beheer en orkestratie mogelijk te maken, is een nieuwe architectuur nodig van losjes gekoppelde beheer- en orkestratie-elementen, met open en programmeerbare interfaces die de communicatie tussen die elementen en de NIF's in de gedistribueerde netwerksegmenten mogelijk maken. Aldus volgen de werkzaamheden van normalisatie-instellingen, zoals het Europees Normalisatie-instituut voor Telecommunicatie (ETSI), in de twee werkgroepen: Zero-touch network and Service Management (ZSM), en Experiential Networked Intelligence (ENI), draagt dit onderzoek bij tot het versnellen van de automatische uitvoering van MANO-operaties door de intelligentie te injecteren in verschillende netwerksegmenten (bv. edge services/EdgeApps, orchestrators, platformbeheerders, radio en kernnetwerk), wat zelfs verder gaat dan het toepassingsgebied van de huidige normalisatieactiviteiten in deze twee werkgroepen. Dit onderzoek opent het potentieel om de standaardisatieactiviteiten uit te breiden en zo bij te dragen tot de standaardisatie-instanties.

Een van de belangrijkste componenten van dit onderzoek is de toepasbaarheid op de volgende generatie reële netwerken en systemen, aangezien het is gedreven door een actief engagement met de industrie, wat resulteerde in een sterke samenwerking met verschillende belangrijke partners uit de 5G-, automobiel- en transportlogistiekindustrieën. Bovendien heeft het proefschrift bijgedragen aan de volgende Europese projecten: H2020 5G-CARMEN, H2020

DAEMON, H2020 VITAL-5G, H2020 5G-BLUEPRINT, en H2020 FED4FIRE+, resulterend in vijf journal papers (gepubliceerd en ingediend), 12 publicaties in conference proceedings (full, work-in-progress, en demo papers; gepubliceerd en ingediend), en een boekhoofdstuk, als eerste auteur, alsmede drie tijdschriften en zes conference papers, als co-auteur (gepubliceerd en ingediend).

Acknowledgements

I would like to express my deepest gratitude to my supervisors Prof. dr. Johann M. Marquez-Barja and Prof. dr. Steven Latré, whose support and guidance helped me tremendously during my Ph.D. research. I certainly learned a lot from you, both professionally and personally. You are my role models in the world of research and academia, and I consider myself lucky to have had a chance to work with you and learn from you throughout these four years.

I also thank my doctoral committee and the jury members, who generously provided knowledge and expertise. Thank you for the fruitful discussion during the internal defense, and for all suggestions that have helped me towards improving the overall outlook and preparing the final version of the thesis. I appreciate your valuable feedback, and your effort and time invested in the final stages of my Ph.D. trajectory.

This endeavor is certainly enriched by generous support from my colleagues from the whole IDLab, IMEC, and the Faculty of Applied Engineering. Thank you for the opportunities to work on the exciting problems and helping me to find solutions and test them in real-life environments. I am grateful to my dear Flexible Networking teammates for making this journey less stressful and much more fun.

Special thanks to partners from various projects that I worked on, as they all played an important role during my Ph.D. research. I would like to extend my sincere thanks to colleagues from NEC Laboratories Europe GmbH. Thank you for the exciting and inspiring technical discussions that resulted in many joint publications.

Last but not least, I would like to thank my family. Thanks to my dear parents for your life-long support and encouragement, and for expressing an interest in my research and work, although it might have sounded abstract and complex from time to time. Finally, thanks to my spouse Irfan for his selfless support, as his belief in me has kept my spirits and motivation high during this process. Thank you for being there every day, and for being my devoted companion on this important journey.

Nina Slamnik-Kriještorac
October 2022, Antwerp

X

Contents

List of Publications

Journal Articles

1. **N. Slamnik-Krieštorac**, H. Kremo, M. Ruffini and J. M. Marquez-Barja, "Sharing Distributed and Heterogeneous Resources toward End-to-End 5G Networks: A Comprehensive Survey and a Taxonomy," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1592-1628, 2020, doi: 10.1109/COMST.2020.3003818, Impact factor: 25.249.

2. **N. Slamnik-Krieštorac**, E. de Britto e Silva, E. Municio; H.C. Carvalho de Resende, S.A. Hadiwardoyo, J.M. Marquez-Barja, "Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context," in *Sensors* 2020, 20, 3852. doi: 10.3390/s20143852, Impact factor: 3.576.

3. **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," in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2021.3118058, Impact factor: 5.577.

4. **N. Slamnik-Krijestorac**, F. Z. Yousaf, G. M. Yilma, R. Halili, M. Liebsch, and J. Marquez-Barja, "Edge-aware Cloud-native Service for Enhancing Back Situation Awareness in 5G-based Vehicular Systems *(Submitted)*," to *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022, Impact factor: 5.978.

5. **N. Slamnik-Krijestorac**, M. Camelo, C. Y. Chang, P. Soto-Arenas, L. Cominardi, D. De Vleeschauwer, S. Latré, and J. Marquez-Barja, "AI-empowered Management and Orchestration of Vehicular Systems in the Beyond 5G era *(Submitted)*," to *IEEE Intelligent Transportation Systems Magazine*, pp. 1–7, 2022, Impact factor: 5.293.

6. E. Zeljković, **N. Slamnik-Krieštorac**, S. Latré and J. M. Marquez-Barja, "ABRAHAM: Machine Learning Backed Proactive Handover Algorithm Using SDN," in *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1522-1536, Dec. 2019, doi: 10.1109/TNSM.2019.2948883, Impact factor: 4.195.

7. R. Halili, F. Z. Yousaf, **N. Slamnik-Krieštorac**, G. M. Yilma, M. Liebsch, R. Berkvens, M. Weyn, "Self-correcting Algorithm for Estimated Time of Arrival of Emergency Responders *(Accepted)*," in *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022, Impact factor: 5.978.

8. G. Miranda, J. Haxhibeqiri, **N. Slamnik-Krieštorac**, Xianjun Jiao, Jeroen Hoebeke, Ingrid Moerman, and Johann M. Marquez-Barja, " The Quality-aware and Vertical-tailored Management of Wireless Time-Sensitive Networks *(Accepted)*," in *IEEE Internet of Things Magazine*, Impact factor: NA.

Conference Proceedings

1. **N. Slamnik-Krieštorac** and J. M. Marquez-Barja, "Demo Abstract: Assessing MANO Performance based on VIM Platforms within MEC Context," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 1338-1339, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162932, Core ranking: A*.

2. **N. Slamnik-Krieštorac**, P. Soto-Arenas, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Realistic Experimentation Environments for Intelligent and Distributed Management and Orchestration (MANO) in 5G and beyond," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 943-944, doi: 10.1109/CCNC49033.2022.9700659, Core ranking: B.

3. **N. Slamnik-Krieštorac**, G. M. Yilma, F. Zarrar Yousaf, M. Liebsch and J. M. Marquez-Barja, "Multi-domain MEC orchestration platform for enhanced Back Situation Awareness," *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1-2, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484632, Core ranking: A*.

4. **N. Slamnik-Krieštorac**, S. Latré and J. M. Marquez-Barja, "An optimized application-context relocation approach for Connected and Automated Mobility (CAM) ," *IEEE 5G for Connected and Automated Mobility (CAM)*, 2021, doi: 10.48550/arXiv.2109.11362, Core ranking: NA.

5. **N. Slamnik-Krieštorac**, M. C. Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Building Realistic Experimentation Environments for AI-enhanced Management and Orchestration (MANO) of 5G and beyond V2X systems," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 437-440, doi: 10.1109/CCNC49033.2022.9700649, Core ranking: B.

6. **N. Slamnik-Krieštorac**, H. C. Carvalho de Resende, C. Donato, S. Latré, R. Riggio and J. Marquez-Barja, "Leveraging Mobile Edge Computing to Improve Vehicular Communications," 2020 *IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 2020, pp. 1-4, doi: 10.1109/CCNC46108.2020.9045698, Core ranking: B.

7. **N. Slamnik-Krieštorac** and J. M. Marquez-Barja, "Unraveling Edge-based in-vehicle infotainment using the Smart Highway testbed," 2021 *IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1-4, doi: 10.1109/CCNC49032.2021.9369622, Core ranking: B.

8. **N. Slamnik-Krieštorac**, M. Peeters, S. Latré and J. M. Marquez-Barja, "Analyzing the impact of VIM systems over the MEC management and orchestration in vehicular

communications," 2020 29th *International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1-6, doi: 10.1109/ICCCN49398.2020.9209636, Core ranking: B.

9. **N. Slamnik-Krieštorac**, G. Landi, J. Brenes, A. Vulpe, G. Suciu, V. Carlan, K. Trichias, I. Kotinas, E. Municio, A. Ropodi, and J. M. Marquez-Barja, "Network Applications (NetApps) as a 5G booster for Transport & Logistics (T&L) Services: The VITAL-5G approach," 2022 *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 279-284, doi: 10.1109/EuCNC/6GSummit54941.2022.9815830, Core ranking: NA.

10. **N. Slamnik-Krieštorac**, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "An ML-driven framework for edge orchestration in a vehicular NFV MANO environment *Accepted*," 2022 *IEEE 20th Annual Consumer Communications & Networking Conference (CCNC)*, 2023, Core ranking: B.

11. **N. Slamnik-Krieštorac**, W. Vandenberghe, R. Kusumakar, K. Kural, M. Klepper, and J. M. Marquez-Barja, "Performance Validation Strategies for 5G-enhanced Transport & Logistics in the 5G Pilot Trials: The 5G-Blueprint Approach *(Accepted)*," *IEEE Future Networks World Forum (FNWF)*, 2022, Core ranking: NA.

12. **N. Slamnik-Krijestorac**, W. Vandenberghe, N. Masoudi-Dione, S. Van Staeyen, L. Xiangyu, R. Kusumakar, and J. M. Marquez-Barja, "On Assessing the Potential of 5G and beyond for Enhancing Automated Barge Control *(Submitted)*," to *IEEE Wireless Communications and Networking Conference (WCNC)*, Core ranking: B.

13. D. Harutyunyan, R. Behravesh and **N. Slamnik-Krieštorac**, "Cost-efficient Placement and Scaling of 5G Core Network and MEC-enabled Application VNFs," 2021 *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 241-249, Core ranking: B.

14. G. M. Yilma, **N. Slamnik-Krieštorac**, M. Liebsch, A. Francescon, and J. M. Marquez-Barja, "No Limits – Smart Cellular Edges for Cross-Border Continuity of Automotive Services *(Accepted)*," *IEEE Future Networks World Forum (FNWF)*, 2022, Core ranking: NA.

15. R. Halili, F. Z. Yousaf, **N. Slamnik-Krieštorac**, G. M. Yilma, M. Liebsch, E. de Britto e Silva, S.A. Hadiwardoyo, R. Berkvens, M. Weyn, "Leveraging MEC in a 5G System for Enhanced Back Situation Awareness," 2020 IEEE 45th Conference on Local Computer Networks (LCN), 2020, pp. 309-320, doi: 10.1109/LCN48667.2020.9314838, Core ranking: B.

16. M. Camelo, L. Cominardi, M. Gramaglia, M. Fiore, A. Garcia-Saavedra, L. Fuentes, D. De Vleeschauwer, P. Soto-Arenas, **N. Slamnik-Krieštorac**, J. Ballesteros, C. Y. Chang, G. Baldoni, J. M. Marquez-Barja, P. Hellinckx, S. Latré, "Requirements and Specifications for the Orchestration of Network Intelligence in 6G," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 1-9, doi: 10.1109/CCNC49033.2022.9700729, Core ranking: B.

17. E. Bërdufi, **N. Slamnik-Krieštorac**, and J. M. Marquez-Barja, "Leveraging on Network Slicing to Enable and Enhance IoT-based e-Health services," In Proceedings of

the 2022 ACM Conference on Information Technology for Social Good (GoodIT '22). Association for Computing Machinery, New York, NY, USA, 431–436. https://doi.org/10.1145/3524458.3548490, Core ranking: NA.

18. B. Ayaz, **N. Slamnik-Kriještorac**, and J. M. Marquez-Barja, "Data Management Platform For Smart Orchestration of Decentralized and Heterogeneous Vehicular Edge Networks," In Proceedings of the 2022 ACM Conference on Information Technology for Social Good (GoodIT '22). Association for Computing Machinery, New York, NY, USA, 118–124. https://doi.org/10.1145/3524458.3547254, Core ranking: NA.

Books

1. N. Slamnik-Kriještorac, J. M. Marquez-Barja, "Mobile edge computing in Internet of Unmanned Things (IoUT) *(Submitted)*," in Emerging Technologies for Internet of Unmanned Things (IoUT) and Mission-based Networking, Springer.

List of Figures

List of Tables

Acronyms

EMS Emergency Management System. 40, 41

OSM Open Source MANO. 31–33, 43, 51, 59, 76, 78, 91, 182

RAN Radio Access Network. 12, 13, 48, 49, 220, 228, 240, 241, 244, 246, 247, 255, 256, 258, 259, 263, 268

ULCL UpLink CLassifier. 40

eMBB enhanced Mobile Broadband. 11, 14, 111, 115, 117, 223, 245, 248, 263

3GPP 3rd Generation Partnership Project. 34, 38, 43, 84, 86, 87, 158, 159, 164–166, 246, 250, 255

5G NORMA 5G NOvel Radio Multiservice adaptive network Architecture. 246

5G NR 5G New Radio. 42

ACN Avoid Close Neighbors. 257

AF Application Function. 87

AGV Automated Guided Vehicle. 115

AI Artificial Intelligence. xxi, 12–15, 18, 21, 37, 43–45, 116, 119–121, 148–150, 152, 153, 156–158, 160–163, 179–181, 184–186, 228

AMF Access and Mobility Management Function. 154

AMQP Advanced Message Queueing Protocol. 61

AP Access Point. 247

API Application Programming Interface. 13, 37, 38, 102, 119, 120, 159, 169

ASE Auto Scaling Engine. 72

AWS Amazon Web Services. 59, 75–78

AWS EC2 AWS Elastic Compute Cloud. 32, 77

B2B Business to Business. 235

B2B2C Business to Business to Consumers. 235

B2C Business to Consumers. 235

B5G Beyond 5G. xxi, 148, 149, 154, 156, 157, 184

BBF Broadband Forum. 248

BBU Baseband Unit. 249, 258, 259

BSA Back-situation Awareness. 20, 37, 39, 42, 121–143, 172, 178, 179, 184

BSC Base Station Controller. 241, 255

BTS Base Transceiver Station. 220, 240, 241, 246, 248, 255, 256, 258, 259, 263, 267, 269

C-ITS Cooperative Intelligent Transport System. 121, 122, 124–129, 135, 139

C-ITS Cooperative Intelligent Transport System. 163

C-RAN Centralized Radio Access Network. 233, 245, 249, 255, 258, 259

CAM Cooperative Awareness Message. 121, 122, 124–127, 129, 132, 134–143, 165, 184

CapEx Capital Expenditure. 127, 227, 236, 248, 250, 256

CCAM Cooperative, Connected, and Automated Mobility. xx, 20, 27, 33, 83, 85, 86, 94, 98, 99, 108, 109, 183

CDF Cumulative Distribution Function. 139

CDN Content Delivery Network. 49

CDNaaS CDN as a Service. 66

CFS Customer Facing Service. 126

CI/CD Continuous Integration and Continuous Delivery. 38

CLI Command Line Interface. 65

CM Configuration Management. 40

CNI Container Networking Interface. 129, 143

CO Central Office. 240

CoAP Constrained Application Protocol. 28

CP Control Plane. 159

CPRI Common Public Radio Interface. 258

CPU Central Processing Unit. 103, 105–108

CR Cognitive Radio. 225, 226, 230, 231, 252, 257, 258

CRI Control-related Information. 59

CSA Constant Spectrum Allocation. 257

CSP Communication Service Provider. 127

D-RAN Distributed Radio Access Network. 255

D2D Device to Device. 226, 252, 257, 258

DAD Dynamic Alternate Direction. 257

DAS Distributed Antenna System. 256

DB Database. 125, 126, 128

DENM Decentralized Environmental Notification Message. 122, 125–127, 129, 132, 134–136, 139–141

DL Downlink. 253, 258, 260, 264, 269

DQN Deep Q-network. 44, 45

DRAM Dynamic Resource Allocation Method. 44

DRI Data-related information. 59

DSA Dynamic Spectrum Allocation. 226

DSRC Dedicated Short Range Communication. 28, 35, 36

EAC Edge Application Client. 165

EAS Edge Application Server. 165

EC-RAN Enhanced Cloud RAN. 250, 259

ECS Edge Configuration Server. 165

EdgeApp Edge Network Application. xiii, xix, xx, 12, 13, 15–21, 27, 37, 38, 40, 45, 66, 79, 80, 109–121, 144, 145, 148, 162, 164, 171, 179–181, 183, 186, 270

EEC Edge Enabler Client. 165

EES Edge Enabler Server. 165, 166

ELBA Efficient Load Balancing Algorithm. 44

EM Equipment Manufacturer. xxi, 221

EMA Emergency Management Authority. 121, 126, 134

EMC Electromagnetic Compatibility. 264

EMS Element Management System. 41

EmV Emergency Vehicle. 20, 41, 42, 121–124, 129, 132, 134, 135, 137–144, 183, 184

ENI Experiential Networked Intelligence. 44, 186

EON Elastic Optical Network. 227, 238, 251, 269

EPC Evolved Packet Core. 241, 244, 255

ERA Efficient Resource Allocation. 44

ESCE Equally Spread Current Execution. 44

ETA Estimated Time of Arrival. 121–128, 134–142, 144, 183, 184

ETSI European Telecommunications Standards Institute. 13, 32, 33, 43, 44, 53, 84, 112, 113, 152, 166, 174, 186, 246

eUTRAN Evolved Universal Terrestrial RAN. 255

FA Federation Agent. 34

FBS Femto BTS. 260, 267

FCC U.S. Federal Communication Commission. 230, 260, 269

FCN Future Communication Network. xxii, 219, 224, 225, 227, 228, 231, 234, 240–244, 247, 248, 250, 253, 255, 256, 260, 262, 266–270

FDD Frequency Division Duplex. 260, 263

FDIO Fast Data Input Output. 129

FDP-MAC Frequency and Distance-based Priority MAC. 42

FL Federated Learning. 156

FM Federation Manager. 34

FOFSA Feedback-Based Optimized Fuzzy Scheduling Algorithm. 44

GGSN Gateway GPRS Support Node. 241

GM Group Management. 40

GMSC Gateway Mobile Switching Center. 241

GNSS Global Navigation Satellite System. 119

GPP General Purpose Processor. 258

GPRFCA Gaussian Process Regression for Fog–Cloud Allocation. 44

GPS Global Positioning System. 121, 124, 137, 138

GPSI Generic Public Subscription Identifier. 91

gRPC Google Remote Procedure Calls. 84

GUI Graphical User Interface. 65

GWCN Gateway Core Network. 250, 256

HCA Hill Climbing Algorithm. 44

HD High-definition. 117

HLR Home Location Register. 241

HOT Heat Orchestration Template. 64

HSS Home Subscriber Server. 241

HTTP HyperText Transfer Procotol. 28, 120

IBFD In-Band Full Duplex. 226, 252, 257

IM Identity Management. 40

InP Infrastructure Provider. xxi, 19, 220, 221, 235, 240, 245, 248, 256, 259, 263

INT Inband Network Telemetry. 247

IoT Internet of Things. 13, 42, 109, 219, 220, 223, 225–228, 231, 232, 235, 241, 242, 245, 247, 248, 252, 253, 256–258, 260, 262–265, 267–270

IP Internet Protocol. 237

IPTV Internet Protocol Television. 48

ISG Industry Specification Group. 13, 44

ISP Internet Service Provider. 245

ITS-G5 Intelligent Transportation System. 132, 184

JSON JavaScript Object Notation. 169

K8s Kubernetes. 38, 91, 92, 162

KM Key Management. 40

KPI Key Performance Indicator. 13, 15, 48, 50, 83, 84, 92, 103, 117, 123, 134, 150, 151, 154, 159, 160, 163, 164, 170, 172, 173, 186, 236, 237, 242, 243, 248, 250, 251, 253, 267, 269

LAA License Assisted Access. 257

LADN Local Area Data Network. 165

LCM Life-cycle Management. 82, 86, 129

LM Location Management. 40

LSA Licensed Shared Access. 257

LSTM Long Short-Term Memory. 165, 168, 179

LTE Long-Term Evolution. 226, 232, 252, 263

LTE-U LTE-Unlicensed. 226, 252, 257, 258

LWA LTE-Wi-Fi Aggregation. 257

LXC Linux Containers. 69, 163

M2M Machine-to-Machine. 11, 223

MAC Medium Access Control. 230, 237, 247, 248

MAESTRO MI-enhAnced Edge Service oRchestration. 170

MANO Management and Orchestration. xiii, xxi, 12, 13, 17, 19–21, 24, 25, 30–34, 38, 43, 44, 49, 75, 76, 78, 79, 86, 148–153, 156–163, 166, 170, 171, 178, 182, 184–186, 246

MAPE-K Monitor-Analyze-Plan-Execute-Knowledge. 152

MARL Multi-Agent Reinforcement Learning. 156

MBS Macro BTS. 260, 267

MBSS Middleware Based Server System. 233, 252, 260

MCDM Multi-Criteria Decision Making. 165, 168, 170, 173–175, 178, 180

MDO Multi-Domain Orchestrator. 30

MEAO MEC Application Orchestrator. 54, 86, 129

MEC Multi-Access Edge Computing. 12–15, 20, 27–30, 32, 33, 42, 43, 48, 49, 54, 55, 76, 82, 83, 86, 91, 93, 96, 98, 121–124, 126–129, 132–134, 136, 138–144, 148, 150, 152, 162–165, 167–170, 172, 177, 181–184, 233, 242, 243

MILP Mixed Integer Linear Programming. 250, 252

MIMO Multiple Input Multiple Output. 231

ML Machine Learning. xxi, 12, 13, 15, 18, 21, 37, 43–45, 116, 119–121, 148–150, 152–154, 156–158, 160–163, 170, 171, 173, 175, 178–181, 184–186

MLA Management Level Agreement. 84, 87, 89, 94–96, 98, 103, 106, 108, 109

MME Mobility Management Entity. 256

mMTC massive Machine Type Communication. 11, 14, 111, 115, 117, 223, 245, 248

mmWave milimeter wave. 232, 250, 252, 263, 268

MNO Mobile Network Operator. 19, 34, 35, 85, 86, 95, 129, 148, 150, 162, 219, 235, 248, 259, 263

MOCN Multi-Operator Core Network. 250, 255

MORAN Multi-Operator Radio Access Network. 255

MQTT Message Queueing Telemetry Transport. 28

MSE Mean Squared Error. 15, 177, 178

MVNO Mobile Virtual Network Operator. 220, 235, 248, 256, 259, 263

NB-IoT Narrowband Internet of Things. 226

NeBV Network-based Virtualization. 245, 257, 260

NFV Network Function Virtualization. xiii, xxi, 11–15, 18, 21, 31–34, 43, 44, 48–50, 53, 59, 82, 84, 94, 112, 121, 148–153, 159, 166, 171, 174, 178, 184, 223, 228, 245–247, 253, 268, 270

NFV-LO NFV Local Orchestrator. 86, 93–96, 98–100

NFV-SO NFV Service Orchestrator. 86, 91, 94, 96, 98, 100

NFVI NFV Infrastructure. 32, 53, 54, 83, 86, 165, 170, 182

NFVO NFV Orchestrator. 44, 53, 54

NI Network Intelligence. xxi, 148, 157–159, 185

NIF Network Intelligence Function. xxi, 148, 152–155, 157–159, 161–163, 184–186

NISP National Industrial Symbiosis Program. 222

NoBV Node-based Virtualization. 245, 257, 260

NOMA Non-Orthogonal Multiple Access. 29, 226, 252, 257, 258

NR New Radio. 12, 14, 232, 252, 264

NRM Network Resource Management. 40

NSA Non-Standalone. 38

NSD Network Service Descriptor. 30, 43, 64, 75–77, 89, 112

NSE Network Slicing Engine. 72

NTIA National Telecommunications and Information Administration. 231

OBU On Board Unit. 49, 141, 163

ODN Optical Distribution Network. 240

OEO Optical-Electrical-Optical. 240, 256

OID Overall Instantiation Delay. 67, 72, 76–78, 182

OL Online Learning. 154, 157

OLT Optical Line Terminal. 240, 245

ONAP Open Network Automation Platform. 31, 33, 34, 38, 43, 60

ONF Optical Networking Foundation. 227

ONU Optical Network Unit. 240, 245

OODA Observe-Orient-Decide-Act. 152

OpEx Operational Expenditure. 127, 227, 248, 250, 256

OTD Overall Termination Delay. 67, 77, 78, 182

OTT Over The Top. 259

P2P Point-to-Point. 230

PAYG Pay As You Go. 235

PBSA Priority based Resource Allocation. 44

PDN Packet Data Network. 87

PGW Packet data network Gateway. 241

PKI Public Key Infrastructure. 264

PoC Poof-of-Concept. xxi, 21, 161, 163–165, 169, 170, 172, 177, 184

PON Passive Optical Network. 231, 233, 240, 245, 256

PoPs Points of Presence. 34

PPDR Public Protection and Disaster Relief. 38, 231, 232

QoE Quality of Experience. 32, 43, 48, 54, 55, 154, 182, 186, 251, 252, 266–268

QoS Quality of Service. 11, 12, 14, 32, 40, 43, 48, 54, 55, 79, 80, 83, 89, 123, 154, 159, 164, 165, 167, 182, 183, 186, 223, 246–248, 252, 255, 259, 266–268, 270

QoT Quality of Transmission. 251

RAM Random-Access Memory. 103, 106

RBF Radial Basis Function. 177

REST REpresentational State Transfer. 33, 84, 91, 102, 120

RF Radio Frequency. 242, 256, 258

RL Reinforcement Learning. 155, 156

RNC Radio Network Controller. 241, 255, 264

RNIS Radio Network Information Service. 86

ROADM Reconfigurable Optical Add/Drop Multiplexers. 227

ROI Return of Investments. 250

RR Round Robin. 44

RRH Remote Radio Head. 249, 258, 259

RSDA Remote Sync Differential Algorithm. 44

RSU Roadside Unit. 20, 41, 132, 162, 170, 177

RTT Round Trip Time. 169

SA Standalone. 40

SDN Software Defined Networking. 11–15, 21, 43, 48, 50, 148, 223, 227, 232, 233, 241, 244–247, 249, 253, 256, 259, 268–270

SDO Software Defined Optics. 232

SDR Software Defined Radio. 230, 232, 244, 249

SEAL Service Enabler Architecture Layer. 38–40, 159

SFC Service Function Chain. 48, 55, 67, 77

SGSN Serving GPRS Support Node. 241

SGW Serving Gateway. 241

SJF Shortest Job First. 44

SL Supervised Learning. 154

SLA Service Level Agreement. 154, 230

SP Service Provider. 219, 220, 235, 245, 248

SRLG Shared Risk Link. 239

SSC Service and Session Continuity. 84, 87

SVR Support Vector Regression. 170, 173, 174, 177, 178, 180

SWS Sub-Wavelength Sharing. 240

T&L Transport & Logistics. 17, 20, 24, 111–113, 115–117, 144, 183

TCCA The Critical Communications Association. 231

TDD Time Divison Duplex. 260, 263, 264

TIP Telecom Infra Project. 227

TML Tiny ML. 157

ToA Time of Arrival. 41

TOPSIS The Technique for Order of Preference. 165, 170, 173, 174

TOSCA Topology and Orchestration Specification for Cloud Applications. 64

TSA Tabu Search Algorithm. 44

TVWS TV White Space. 260

UE User Equipment. 13, 14, 39, 45, 220, 259

UL Uplink. 154, 252, 253, 258, 263, 264, 269

UP User Plane. 159

UPF User Plane Function. 40, 87, 167

URLLC Ultra-Reliable and Low-latency Communication. 75, 121

uRLLC ultra-Reliable Low Latency Communication. 11, 14, 111, 115, 117, 223, 245, 247, 248, 263

V2I Vehicle-to-Infrastructure. 41, 123

V2N Vehicle-to-Network. 36, 123

V2V Vehicle-to-Vehicle. 41, 123

V2X Vehicle-to-Everything. xxi, 18, 20, 27, 29, 36, 39, 41, 42, 124, 127–129, 131, 132, 148–160, 162, 170–172, 174, 177–179, 184

VAE V2X Application Enabler. 159

VAL Vertical Application Layer. 39, 40

VAS Value-added Service. 86

vCDN virtual Content Delivery Network. xxi, 161, 163, 164

VHF Very High Frequency. 229

VIM Virtual Infrastructure Manager. 32, 34, 51, 53, 57, 66, 75–79, 182

VLR Visitor Location Register. 241

VM Virtual Machine. 32, 55, 56, 76–78, 182

VNF Virtual Network Function. xix, 11, 14–16, 19, 20, 33, 34, 44, 48, 53–56, 77, 78, 82, 113, 144, 150, 157, 243

VNFD VNF descriptor. 30, 43, 64, 75, 76, 89, 112

VNFM VNF Manager. 34, 53

VNO Virtual Network Operator. 240, 245, 249, 262

VPN Virtual Private Network. 237

WDM Wavelength Divison Multiplexing. 231

WP Waypoint. 124–126

WSN Wireless Sensor Network. 231, 233, 241, 245, 252, 257, 260

YANG Yet Another Next Generation. 64

ZSM Zero-touch Network and Service Management. 44, 186

Chapter 1

Introduction

The digital technologies have been constantly improving the functioning of societies over the past few decades, thereby transforming numerous aspects of a modern life, starting from safety and transportation, all the way to our health and well-being. In particular, digitalization enables an extensive set of technological devices such as sensors, phones, vehicles, and facilities, to become more efficient, cost-effective, and significantly more versatile, whereas an enormous amount of data is being collected all around the digital ecosystem in an instantaneous manner. However, a prerequisite for bringing the aforementioned to reality is a ubiquitous connectivity of all these devices, making them able to connect and to exchange information via network. In particular, there is a prediction in Cisco Forecast and Trends paper [3] that an ever-increasing number of devices that are wirelessly connected to the Internet will reach approximately 12.3 billion in a less than a year from now. As a consequence, such growth unavoidably leads to tremendous increase in network traffic, i.e., service requests for applications like video, Machine-to-Machine (M2M) communications, and interactive gaming, stretching to more advanced services for vehicular communications, transport and logistics, and e-health systems, which all-together impose stringent requirements on latency, reliability, and bandwidth.

In the 5G and beyond community, all the above applications fall into the three main areas: massive Machine Type Communication (mMTC), ultra-Reliable Low Latency Communication (uRLLC), and enhanced Mobile Broadband (eMBB) [4], while applications are being mapped to these three categories depending on the highly specific and stringent Quality of Service (QoS) requirements they impose. For the network operators, these QoS requirements are then tied to provisioning of different network resources, where the excessive growth in service requests becomes a heavy technological and economical burden. To accommodate new digital systems and services, networks need to be programmable, thus bringing significantly more flexibility and adjustability to the way how resources are being managed and allocated. Such programmability is achieved through network softwarization, which is based on the two main pillars, i.e., the NFV and Software Defined Networking (SDN). In particular, NFV virtualizes network resources from the underlying physical infrastructure through the concepts of abstraction and isolation [5, 6, 7], and enables dynamic creation of VNFs that are further used as building blocks for designing complex and robust network services. On the other hand, SDN is in charge of programming the way those VNFs are connected to each other over the network, thereby decoupling the network control from the data plane. The main advantages of both NFV and SDN are seen in the opportunities for

achieving an effective, flexible, and dynamic management of resources in modern computing environments.

The 5G ecosystem is one of such environments, as it spans a wide variety of resources from the 5G New Radio (NR) and other types of Radio Access Networks (RANs), through edge, transport, and core network, to the data network, forming a robust software-based system that relies on both SDN and NFV to achieve network programmability. However, the virtualization of 5G network (e.g., core and edge networks, and partially the radio side) results in an extensive pool of diverse and heterogeneous resources, and one of the main challenges is to properly manage and orchestrate such resources that are distributed across the overall 5G ecosystem. Hence, the role of the MANO is to i) identify the resource needs for a service running on the virtualized network infrastructure, and ii) to address those needs in way that is dynamic and does not affect the QoS and service continuity, by performing proactive service reconfigurations (e.g., scaling, migration, and service teardown, which are known as MANO operations). To this end, in this PhD research, we have been investigating *service and resource management and orchestration* in *collaborative and distributed Multi-Access Edge Computing (MEC) environments* to enable *openness* and *programmability* of 5G and beyond ecosystems. The word **'collaborative'** refers to collaboration between different edge/MEC orchestrators in performing orchestration of services and EdgeApps deployed at the 5G edges, which belong to different administrative (e.g., two mobile operators) and/or different technological domains (e.g., the same operator, but different technologies used, such as OpenStack and Kubernetes). Since 5G is mainly designed to boost the operation of verticals (automotive, transport & logistics, e-health, etc.), vertical services that are built to deliver new use cases for those verticals can be deployed at the network edge, in order to experience lower latency and higher bandwidth. As edges can be spawned anywhere in the network infrastructure (e.g., collocated with base stations, roadside units on the highways, or small data centers in labs), depending on the location of their mobile users, vertical services and their constituting EdgeApps can be deployed in a distributed manner, stretching over multiple edge platforms or being migrated from one to another depending on the service performance. Thus, such distributed infrastructure resources and services/EdgeApps need to be orchestrated by **distributed orchestration elements**, i.e., edge orchestrators that are collaborating with each other while being distributed across the 5G edges. Leveraging on NFV and SDN to achieve network programmability, **edge networks** are enabling **virtualized** and **programmable** service chains, i.e., vertical services and EdgeApps that are loosely coupled via **open** interfaces, which can be efficiently reconfigured based on the decisions made by orchestrators.

Therefore, the main objective of this research is to leverage on 5G, MEC, and AI/ML, to achieve an efficient and automated management and orchestration of services and resources across different technological/administrative edge domains, which enables the low-latency-aware edge placement and seamless migration of programmable edge services and EdgeApps. This objective has been achieved through the four main contributions summarized and briefly discussed in Section 1.3, which are further elaborated in different chapters of this thesis, as described in Section 1.4.

1.1 Background

In the scope of this thesis, we have worked with several fundamental technologies, such as MEC, NFV, SDN, 5G, and AI/ML. Thus, in this section, we provide more insights into the main characteristics of those technologies, as an introduction for Section 1.2, which elaborates on our motivation for researching collaborative and distributed MANO of open programmable and virtualized edge networks. In addition, as the research on this thesis reflects a strong applied engineering component, throughout the thesis we introduce and define various Key Performance Indicators (KPIs) to test and validate the performance of collaborative edge orchestration of services/EdgeApps, as well as the service/EdgeApp performance. Thus, in Section 1.1.2, we provide the list of the most common KPIs that are used in this thesis (from Chapter 3 onwards).

1.1.1 Technologies

Multi-Access Edge Computing Edge computing emerges as an indispensable component of 5G ecosystems, as it brings both the computing resources and the computing capabilities to the network edge, i.e., significantly closer to the end users (e.g., vehicles, pedestrians, vessels, or Internet of Things (IoT) devices), thus, building service-tailored edge clouds that can be accessed with a decreased latency, an improved bandwidth, and a significantly decreased backhaul network utilization [8]. By bringing a powerful and on-demand accessible cloud computing system to the network edge, MEC is considered as a key component of 5G ecosystems, which enables ubiquitous ultra-reliable and low-latency connectivity to distributed services [9, 8], while entailing computing engines that are located either within the RAN of mobile operators, or their transport network. Therefore, the main purpose of designing and deploying MEC platforms for 5G ecosystems is to further reduce the latency in accessing services that were previously deployed on top of the distant locations in the communication systems, such as clouds or private data centers, while offloading heavy computing tasks from a User Equipment (UE), and eliminating the need for running complex and resource consuming tasks at the user side. To standardize a comprehensive architecture of MEC systems and a set of Application Programming Interfaces (APIs) for essential MEC interfaces, the European Telecommunications Standards Institute (ETSI) created an Industry Specification Group (ISG), i.e., ISG MEC [9, 10]. In this thesis, we have leveraged on such a standardized architecture, and used it as a baseline for creating more complex and comprehensive edges that collaborate through the flexible and efficient orchestration layer towards building an agile virtualized environment for edge services of various verticals (e.g., automotive, and transport & logistics).

Network Function Virtualization Regardless of the domain it applies to, virtualization can be defined as abstraction, isolation, and flexible sharing of heterogeneous resources among multiple actors (network operators or users) in both wireless [6, 7] and optical domains [11, 12]. When it comes to virtualizing network functions, the NFV is decoupling those functions from a dedicated hardware in mobile communication systems [13], creating the means for their efficient deployment and management. In particular, 5G networks introduce virtualization and softwarization concepts i) to abstract the complexity of communication

systems, detaching services from an underlying physical network infrastructure [5], and ii) to manage resources more efficiently towards improving QoS levels perceived by end users [14, 15]. Thus, the NFV can be considered as a technique that enables network services to observe and use network resources independently from the underlying physical infrastructure [5], thus, creating the possibility to use network resources in a more scalable and customizable way, as the resource utilization can be mapped with the service requirements, gaining significant reduction in time and resources for network deployment and operation [16].

Software Defined Networking Side by side with the NFV, SDN is one of the main pillars of 5G communication systems, and it is defined as an emerging programmable architecture that decouples network control from data plane. The traditional network devices contain both functionalities, i.e., they make decisions about traffic processing and flow (control plane), and they forward the traffic from one interface to the other (data plane) [17], making it complex to manage the distributed flows and conflicting decisions made by different devices. Given that NFV virtualizes network functions, deploying them as VNFs on top of the distributed computing units, the importance of SDN for mobile communication systems lays in the effective and dynamic resource and processing management of those modern computing environments, which is imposed by control and data plane separation [18]. In particular, the centralized control in 5G networks, brought by SDN, supports and enables service-oriented operation, which is dynamic, easily manageable, cost-effective, and customizable to the emerging and 5G-specific applications (i.e., eMBB, mMTC and uRLLC) [7]. The typical SDN architecture consists of the three main layers: i) the application layer, combining all the applications and network functions ii) the control layer, which is the main component of the SDN system as it manages the policies and the traffic flow for the network, and iii) the infrastructure layer, which consists of all physical switches in the network [17].

5G systems The fifth generation of mobile communication systems is the latest step in the evolution of networks, offering unprecedented QoS for the usual telecommunication operations served by previous generations, but also for various new use cases within many industry verticals (e.g., automotive, transport and logistics, and e-Health), which could have not been possible before. With an explosive growth of data traffic, a significant increase of connected devices, and an advent of new services and applications in various vertical industries, 5G systems are designed to support 100 Mbps-1 Gbps data rate, 1 ms radio transmission latency, and 1 million connections per square kilometer [19]. The 5G ecosystems stretch over the UE, NR, distributed edges, 5G core, and cloud environments, thereby operating at sub-6 GHz frequency range (especially at 3.5-4.2 GHz), as well as at the millimeter wave range (26, 28, 38, and 60 GHz). The main pillars of 5G technology are i) NFV, which virtualizes both network functions, and the functions constituting application services designed for vertical industries, ii) SDN, which intelligently steers the traffic flows through the decentralized network devices, and iii) MEC, which deploys network and application functions closer to the end users, i.e., at the network edge, thereby decreasing the latency and improving bandwidth utilization.

Artificial Intelligence and Machine Learning AI is the computer science field that simulates the processes of human intelligence on computer systems, thereby applying the prin-

ciples of data acquisition and processing, information resolving, as well as deducting conclusions, on the machines [20]. ML is an AI subset that focuses on enabling machines to automatically learn new behaviors based on the past experiences. Given the success of AI/ML in fields such as image and video processing, forecasting, and anomaly detection, there is an increasing demand to explore the possibilities of applying the same or similar mechanisms to solving complex network problems.

1.1.2 Key Performance Indicators

In this thesis, we study and propose new methodologies for both the orchestration systems, and the orchestrated services/EdgeApps, thereby providing guidelines on how these services and EdgeApps should be designed, developed, and orchestrated. Thus, to analyze and validate the performance of both orchestration systems and services/EdgeApps, it is important to make a clear distinction of different groups of KPIs that need to be measured and studied to understand the behavior of orchestrators and services/EdgeApps, and to understand the impact orchestration has on those services/EdgeApps.

Thus, in Table 1.1, we list and define the KPIs that we used in this thesis, grouping them into: i) Orchestration-related KPIs, which measure the performance of orchestration systems, and ii) Service/EdgeApp-related KPIs, which are experienced by the end user and reflect on the performance of a particular edge service/EdgeApp. Each of these KPIs is defined and measured in a specific context, and as such presented in different chapters of this thesis (Table 1.1), but this overview helps the reader to understand the importance of those KPIs and justification of their usage in experimentation setups and performance evaluations. In addition to orchestration-related and service-related KPIs, in Chapter 6 we measured standard deviation and Mean Squared Error (MSE) to evaluate performance of particular AI/ML models that have been leveraged by orchestrators to improve their decision-making for edge orchestration operations.

1.2 Motivation

In this section, we briefly reflect on the main motivation aspects for working on different contributions within this PhD thesis.

As the NFV and SDN are one of the main technology enablers in the 5G ecosystems, creating a clear separation between control and data plane for virtualized network functions, they also provide the means for an agile life-cycle management of the VNFs associated to various edge services and EdgeApps deployed over the same 5G network infrastructure. Taking into account the latency and reliability requirements for modern vehicular communications services, MEC systems are becoming widely popular in delivering a localized access to virtualized services, thereby deploying the vertical services and EdgeApps close to the users (i.e., vehicles). Thus, one of the main motivations for the research we conducted within this PhD thesis was *to leverage on those technology enablers, i.e., NFV, SDN and MEC, and to exploit the opportunities they could bring to the highly challenging application services*, if

Table 1.1: The most common KPIs used for performance evaluations in this thesis.

Type	KPI	Unit	Definition	Chapter
Orchestration	Orchestration latency	[ms]	The time needed for an orchestration operation to be performed by edge/MEC orchestrators. Examples of such orchestration operations are service on-boarding, instantiation, scaling, relocation, and termination. This type of KPI consists of subtypes (on-boarding/instantiation, runtime, and termination), depending on the timing when the orchestration operation is performed, i.e., either before, or during the lifetime of the edge service.	Chapters 3, 4
	On-boarding and instantiation latency	[ms]	The time needed for edge/MEC orchestrators to on-board and instantiate edge service or EdgeApp. In particular, the on-boarding process refers to on-boarding of the application package (e.g., descriptor, Docker container image, or VM image) in all edge nodes within the domains where service or EdgeApp deployment is required. The instantiation latency is the time needed for orchestrators to process the orchestration request and to instantiate services and EdgeApps on the selected edges.	Chapters 3, 4
	Runtime orchestration latency	[ms]	The time needed for the orchestrators to perform any life-cycle management operation, i.e., while the service or EdgeApp instance is running. Some examples of these operations are service or EdgeApp scaling up/down/in/out, relocation/migration, and termination.	Chapters 3, 4
	Termination latency	[ms]	The time needed for orchestrators to process the termination request and to terminate services and EdgeApps. This process includes releasing of the previously occupied network and computing resources.	Chapters 3, 4
	Orchestration load (memory load, CPU load, power consumption)	[%]	Average CPU/memory/power usage during orchestration operations (i.e., processing orchestration requests, performing life-cycle management operations, etc.).	Chapters 3, 4
Service/EdgeApp	End-to-end latency	[ms]	The overall round-trip time for an IP packet, including the time: i) to transfer the packet from the user (e.g., vehicle) over the network to the application service or EdgeApp, ii) to process at the application/EdgeApp level, and iii) to receive back the response from the application/EdgeApp. It includes both communication latency (network impact) and computational latency (application impact).	Chapters 5, 6
	Computational latency	[ms]	The overall time needed for a service or EdgeApp running on the 5G edge to perform its task, i.e., to process the IP packets received from the UE, and to prepare the response, if applicable.	Chapters 5, 6
	Communication latency	[ms]	The time needed for an IP packet to be transferred from UE to N6 interface (5G PPP definition). In particular, user plane latency per 3GPP TR 38.913 definition is the time to successfully deliver an application layer packet/message from the radio protocol L2/L3 Service Data Unit (SDU) entering point to the radio protocol L 2/L3 SDU entering point via the radio interface in both uplink and downlink directions.	Chapters 5, 6
	State update latency	[ms]	The data-plane latency in communication between two peering instances of service or EdgeApps, which are running in different edge domains. Depending on the type of use case, this communication between peering instances improves the situational awareness of vehicular EdgeApps running at the edge, as they can proactively share metadata (location, speed, heading) about the users to which they are connected. This KPI corresponds to the ETSI's context-update time [21].	Chapter 5
	Service load (memory load, CPU load, power consumption)	[%]	Average CPU/memory/power usage during service or EdgeApp runtime.	Chapter 5
	Service reliability	[%]	The probability that a service or a EdgeApp will maintain performance standards for a specific period of time. In particular, it can be calculated as the number of application layer messages that are successfully delivered to a receiver (user, or EdgeApp) within the time required by the respective use case, with respect to the total number of sent messages.	Chapter 6

combined into an integrated resource management and orchestration framework. The design, prototyping, and extensive performance evaluations of such frameworks are extremely important for the future of the software-based networks, as they are an ultimate enabler of innovative use cases that were not possible with the previous generations of mobile network communication systems.

The aforementioned advancements in network programmability and loose coupling between VNFs mitigate the dependency on proprietary hardware equipment and underlying physical network resources by decoupling network functions from infrastructure. Thus, in the beginning of this research we were motivated to exploit the potential of resource sharing in end-to-end 5G networks, which provides incentive to abandon the exclusive possession of network resources and rather opt for sharing. We posed the questions on i) *what and which network resources from 5G ecosystems (both wireless and optical domains) could be shared*, ii) *what can be learned from resource sharing trends in previous generations of communication systems*, and iii) *what are the remaining challenges that still need to be tackled after adopting resource sharing techniques*. As resource sharing is not necessarily part of resource orchestration, we present the outcomes of our research and answers on the previously stated questions in the Appendix A. As follows, we focus further on the motivation aspects for conducting research in the scope collaborative resource and service orchestration, which is the core part of this thesis.

Motivation 1 In the highly agile vehicular environments, cellular systems and in particular 5G could provide vehicles with an extended situational awareness by assisting them in their maneuvering operations. This can be done by deploying the services at the network edge to decrease the end-to-end latency, as such services could collect data not only from one but from multiple vehicles that are even not in the close proximity from each other. Such extended awareness could lead to optimal maneuvering decisions that standalone vehicles cannot make based on their sensors and messages received from the vehicles in their vicinity. Thus, for such use cases, it would make sense to leverage on maneuvering assistance from the 5G network edges with service and EdgeApp deployments, thereby using Uu interface to connect vehicles to those edge services. However, maintaining service continuity in low-latency and high reliable communication with distributed instances of services and EdgeApps at the network edge requires a continuous real-time monitoring and seamless service reconfiguration, as well as relocation of the service, and the user's connection to a more suitable service instance. To achieve such service continuity, *MANO systems need to be effective* in deploying distributed instances of EdgeApps in different decentralized edge domains, and seamless service/EdgeApp reconfiguration and relocation in such highly mobile and resource constrained ecosystem. Thus, due to the *lack of integration and between 5G, MEC, and NFV technologies*, which are standardized by different bodies, *the interplay between these technologies is not sufficiently explored and evaluated when it comes to highly mobile network scenarios such as those created for vehicular communications*. Also, there is an *insufficient support for service continuity in cross-domain edge environments*, as most of the research has been focused on the network handover, and not on the proactive service reconfiguration and relocation based on the users' mobility and infrastructure monitoring data. Our study of the aforementioned challenges, followed by the proposed solutions and mechanisms, is presented in Chapters 3 and 4.

Motivation 2 Concerning the types of edge services and EdgeApps that could benefit from an efficient resource and service management and orchestration, we primarily *focus on the vehicular type of such services as their requirements on the service performance* (latency, bandwidth, service continuity) are strict and the consumers of such services are highly mobile, moving with inherently high speeds, which imposes additional challenges towards the service orchestration and real-time monitoring. The *design of such services is usually challenging and requires to deliver an increased situation awareness*, thereby processing large amounts of data in extremely short periods of time, both from the vehicles and network infrastructure. Also, there is a lack of standardized frameworks for designing and developing such application services. In addition to automotive sector, the Transport & Logistics (T&L) is a major component of modern production and distributed systems, as it significantly contributes to the macroeconomic deployment. However, *the state-of-the-art approaches in performing T&L processes suffer from insufficient automation and process optimization*, which hinders both the efficiency and safety of the T&L operations, and there is still *a knowledge gap between vertical industries and 5G system providers that hinders the benefits of applying 5G EdgeApps to the aforementioned verticals*. Thus, our study of EdgeApps for bringing 5G closer to the vertical industries such as automotive and T&L, and the design guidelines for vertical edge applications decoupled from the underlying physical infrastructure, are presented in Chapter 5.

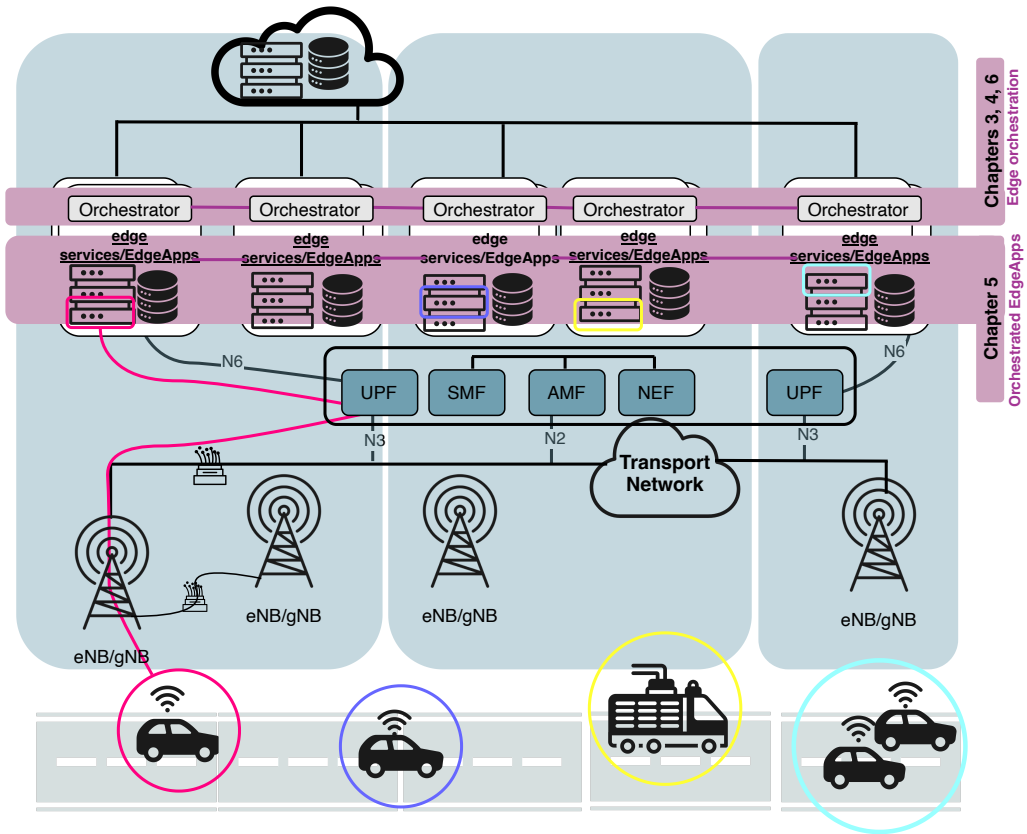


Figure 1.1: The scope of this thesis; 5G ecosystem with distributed orchestration elements and service/EdgeApp deployments across multiple 5G edges.

Motivation 3 Given *the lack of automation in performing management and orchestration operations*, and lack of standardization efforts in automating the NFV operations, we focused on removing the dependencies on the manual interventions in orchestration operations, which usually delay the decision-making executed by orchestrators, thereby risking to significantly affect the service. Thus, we carefully studied the suitability of different AI/ML techniques, such as supervised and unsupervised learning, reinforcement, deep, and federated learning, for making orchestration decisions more intelligent and informed, and we evaluated the impact of AI/ML on edge V2X service relocation in a real-life testbed environment (leveraging on the Smart Highway testbed). The outcomes of our research, i.e., the proposed closed-loop framework for automating edge service orchestration and AI/ML techniques and algorithms that enhance the decision-making process of orchestration operations are presented in Chapter 6.

1.3 Contributions

Here we summarize the main contributions of the thesis, together with the publications relevant for each of the contributions. In particular, Fig. 1.1 illustrates the scope of our contributions in the overall 5G ecosystem, i.e., the edge orchestration concepts and frameworks (Chapters 3, 4, and 6), and the orchestrated EdgeApps (Chapter 5). Furthermore, Fig. 1.2 maps the publications (either published or submitted to journals and conferences) to each of the chapters and the contributions. These main contributions are briefly presented as follows:

1. **Feature and performance evaluation of existing orchestration solutions** (Chapter 3)
   At the beginning of this PhD research, we focused on investigating the end-to-end 5G network resources and service programmability, which has a quite broad perspective as the end-to-end 5G networks entail both wireless and optical domains, altogether with the edge and cloud, as illustrated in Fig. 1.1. In particular, we started from resource sharing, which can be defined as the paradigm that shifts the exclusive ownership of network resources to mutual resource use, thereby enabling service performance improvements and cost savings at the same time (Appendix A). The NFV and SDN are the ultimate enablers of network programmability, and together they are also one of the main pillars of 5G resource sharing, since they virtualize the resources and services, and enable loose coupling of VNFs, making them shareable between Mobile Network Operators (MNOs), InPs, and service developers, among other participants in sharing. Given that such programmability of service function chains opens up the opportunities for achieving an efficient and automated service and resource management, we evolved from a resource sharing perspective to a comprehensive and immensely challenging resource and service management and orchestration of distributed 5G edge networks (Fig. 1.1). Thus, we studied a life-cycle management of 5G services as a closed-loop of the three main intertwined phases: orchestration, control, and monitoring. In all of the three phases, SDN and NFV are applied to bring more flexibility, and programmability to wired and wireless communication networks, and to achieve higher resource utilization, and lower costs. Thus, as a part of the first contribution, we conducted a feature and performance analysis of existing orchestration solutions, thereby experimenting with a testbed setup (Virtual Wall testbed, Ghent, Belgium) to analyze the performance of those MANO solutions, studying their suitability and readiness for orchestrating vehicular services in distributed edge environments. In Fig. 1.2, we list the papers that are published in the scope of the presented contribution.

2. **Resource and service orchestration for Connected Cooperative and Automated Mobility** (Chapter 4)
   Due to the stringent requirements on the ultra-low latency (1-10ms), high-reliability (99,999%), high throughput (up to 20Gbps), and highly efficient network resource consumption, vehicular use cases are pressing the needs for a ubiquitous support form the network and virtualized infrastructure, thereby making automotive industry one of the most challenging and demanding consumer within the 5G ecosystem. Given such demands, the edges of the 5G network seem to be a corresponding candidate for placement of the vehicular services/EdgeApps, thus, making them closer to the

vehicles that are consuming the service, thus enabling service usage with significantly decreased latency and increased throughput. However, as 5G edges are usually not even closely resourceful as remote data centers in the cloud, they require a proper treatment in terms of service and resource management. Another challenge is the inherently high speed of vehicles, which requires from orchestrators to promptly and efficiently reconfigure the distributed deployment of the service, and find a new suitable location based on the mobility and resource demand. Given such research challenges, the work presented in Chapter 5 of the thesis focused on orchestrating this particular type of services, as illustrated in Fig. 1.1, where we investigated and proposed a comprehensive MANO framework for the collaborative orchestration of services for CCAM within such 5G ecosystem. The key objective of this research is to achieve the service continuity for a highly dynamic automotive scenario, through performing associated management and orchestration of these services in distributed edge clouds. The extensive performance evaluation of this orchestration framework for distributed vehicular service deployments has been conducted in various distributed proof-of-concept setups where we utilized the Virtual Wall (Ghent, Belgium) and CityLab (Antwerp, Belgium) testbeds. This particular contribution is presented in Chapter 5 of this thesis, based on the publications listed in Fig. 1.2. In addition, the microservice-based MEC application orchestrator, which is an outcome of this research, is a part of the overarching framework for seamless cross-domain MEC orchestration that we designed and developed within 5G-CARMEN project.

3. **Orchestrated Edge Network Applications (EdgeApps)** (Chapter 5)
   To be able to benefit from 5G technologies in terms of ultra-low latency, high reliability, and extensive throughput, the vertical services need to be properly managed and orchestrated, but their design also needs to be tailored to particular use cases, taking into account vertical service-specific requirements towards 5G. Thus, by applying the cloud native principles and programmability of service function chains to the design and development of vertical services in 5G ecosystems, we have worked on the concept of EdgeApps, as a fundamental building block of the 5G-enhanced automotive and T&L service chains. As illustrated in Fig. 1.1, such EdgeApps are deployed on top of the edge and cloud 5G-enabled infrastructure, and used for creating any complex 5G vertical service by abstracting the underlying 5G network complexity, and thus bridging the knowledge gap between vertical stakeholders, network experts, and application developers. Furthermore, to study the impact of the operations of the collaborative orchestration in distributed edge environments on the EdgeApps built for vehicular use cases, we have been focused on a use case that aims to enhance mission-critical services by leveraging 5G technology, and placing enhanced VNFs or EdgeApps on the 5G network edge. In particular, we designed the Back-situation Awareness (BSA) application service to support Emergency Vehicles (EmVs) on the roads by increasing awareness about them among other civilian vehicles. Thus, the all-encompassing concept of EdgeApps, including a more specific BSA application service, are both presented in the Chapter 5, where we also show and discuss the performance evaluation of the BSA as a 5G V2X use case. The experimentation and performance analysis of the BSA, including its management and orchestration, have been conducted utilizing the Smart Highway testbed, where the Roadside Units (RSUs) installed along the E313 highway (Antwerp, Belgium) served as distributed edge computing nodes. As an outcome of this research, we have created a robust software solution, i.e., a cloud-

native and microservice-based edge application service (i.e., EdgeApp) for addressing back-situation awareness in the real-life environments, which is used and tested with the PoC vehicles on the 5G corridor between Italy and Austria.

4. **Intelligent and automated management and orchestration of services and resources** (Chapter 6)
   One of the grand downsides in management and orchestration solutions is the manual execution of the MANO operations, which might cause large delays in service reconfiguration (e.g., scaling, deployment, termination), thus ultimately affecting the service performance through increased response time (i.e., end-to-end latency), or even causing the service downtime and unavailability. Such consequences become even more severe in the context of previously studied vehicular services, given their stringent service requirements. To this end, in the final stage of this PhD research, we focused on studying the potential of AI/ML techniques to enable automated and highly efficient MANO operations (Fig. 1.1). Some of the traditional approaches for making decisions in service management and orchestration are the human-in-the-loop approach, which is slow and prone to errors, and the closed-loop control that uses rule-based models, which are difficult to design given a large number of parameters that need to be configured and optimized. With such approaches, it becomes hardly feasible to efficiently orchestrate complex and dynamic vehicular environments, and applying AI/ML in combination with NFV and SDN is a promising solution for enabling automation and intelligence that will optimize MANO operations. In the final chapter of this thesis, we carefully studied the gaps in current NFV MANO solutions for efficient orchestration of 5G vehicular edge services, and based on such gaps, proposed an AI/ML-based closed-loop control framework for orchestrating 5G services in an automated and intelligent way. We also identified some specific AI/ML techniques that can alleviate the studied gaps, and also listed the potential implications resulting from applying certain AI/ML techniques. Finally, to test and validate the benefits of applying some of the AI-enhanced algorithms to MANO operations, we have built a realistic proof-of-concept setup using the Smart Highway and Virtual Wall testbeds. Same as for the other contributions, this final contribution is covered in publications shown in Fig. 1.2.

These four main contributions are described in detail within the following publications:

1. **N. Slamnik-Krieštorac**, H. Kremo, M. Ruffini and J. M. Marquez-Barja, "Sharing Distributed and Heterogeneous Resources toward End-to-End 5G Networks: A Comprehensive Survey and a Taxonomy," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1592-1628, 2020, doi: 10.1109/COMST.2020.3003818, Impact factor: 25.249.

2. **N. Slamnik-Krieštorac**, E. de Britto e Silva, E. Municio; H.C. Carvalho de Resende, S.A. Hadiwardoyo, J.M. Marquez-Barja, "Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context," in *Sensors* 2020, 20, 3852. doi: 10.3390/s20143852, Impact factor: 3.576.

3. **N. Slamnik-Krijestorac**, G. M. Yilma, M. Liebsch, F. Z. Yousaf and J. Marquez-Barja, "Collaborative orchestration of multi-domain edges from a Connected, Cooper-

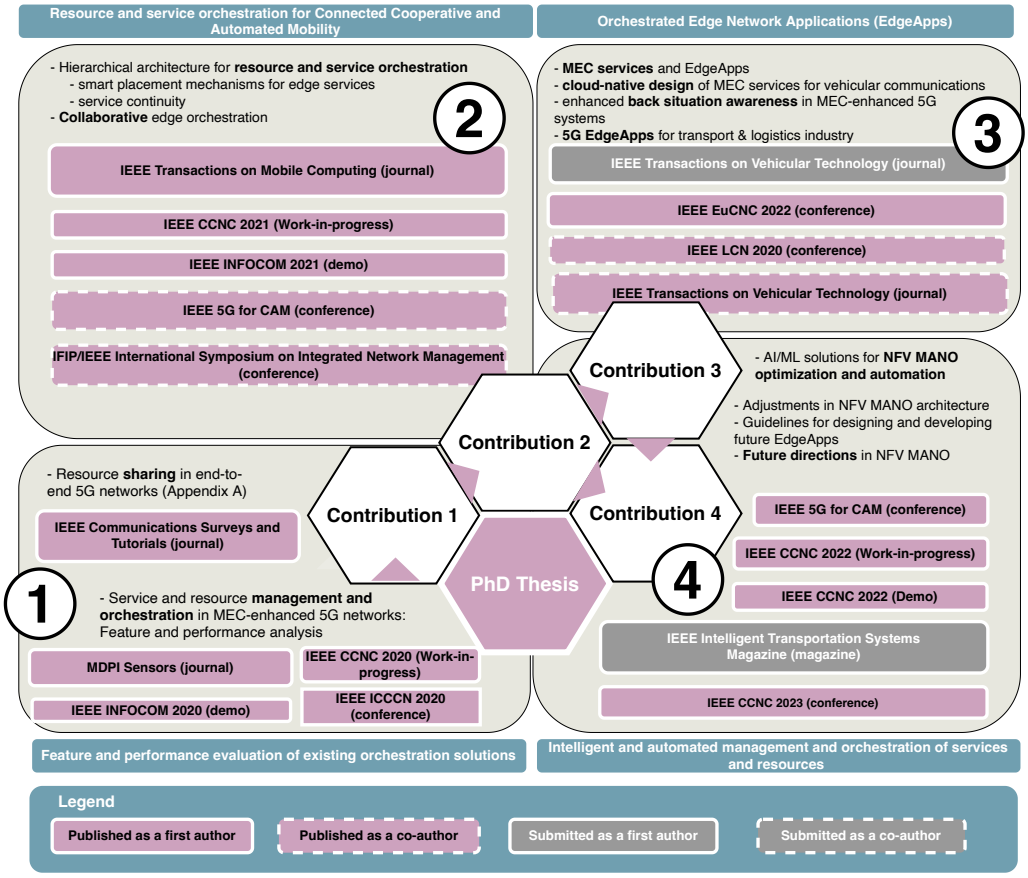


Figure 1.2: Mapping publications to contributions.

ative and Automated Mobility (CCAM) perspective," in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2021.3118058, Impact factor: 5.577.

4. **N. Slamnik-Krijestorac**, F. Z. Yousaf, G. M. Yilma, R. Halili, M. Liebsch, and J. Marquez-Barja, "Edge-aware Cloud-native Service for Enhancing Back Situation Awareness in 5G-based Vehicular Systems *(Submitted)*," to *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022, Impact factor: 5.978.

5. **N. Slamnik-Krijestorac**, M. Camelo, C. Y. Chang, P. Soto-Arenas, L. Cominardi, D. De Vleeschauwer, S. Latré, and J. Marquez-Barja, "AI-empowered Management and Orchestration of Vehicular Systems in the Beyond 5G era *(Submitted)*," to *IEEE Intelligent Transportation Systems Magazine*, pp. 1–7, 2022, Impact factor: 5.293.

6. **N. Slamnik-Kriještorac**, S. Latré and J. M. Marquez-Barja, "An optimized application-context relocation approach for Connected and Automated Mobility (CAM) ," *IEEE 5G for Connected and Automated Mobility (CAM)*, 2021, doi: 10.48550/arXiv.2109.11362, Core ranking: NA.

7. **N. Slamnik-Kriještorac** and J. M. Marquez-Barja, "Demo Abstract: Assessing MANO Performance based on VIM Platforms within MEC Context," *IEEE INFOCOM 2020 -*

IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 1338-1339, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162932, Core ranking: A*.

8. **N. Slamnik-Kriještorac**, P. Soto-Arenas, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Realistic Experimentation Environments for Intelligent and Distributed Management and Orchestration (MANO) in 5G and beyond," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 943-944, doi: 10.1109/CCNC49033.2022.9700659, Core ranking: B.

9. **N. Slamnik-Kriještorac**, G. M. Yilma, F. Zarrar Yousaf, M. Liebsch and J. M. Marquez-Barja, "Multi-domain MEC orchestration platform for enhanced Back Situation Awareness," *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1-2, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484632, Core ranking: A*.

10. **N. Slamnik-Kriještorac**, M. C. Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Building Realistic Experimentation Environments for AI-enhanced Management and Orchestration (MANO) of 5G and beyond V2X systems," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 437-440, doi: 10.1109/CCNC49033.2022.9700649, Core ranking: B.

11. **N. Slamnik-Kriještorac**, H. C. Carvalho de Resende, C. Donato, S. Latré, R. Riggio and J. Marquez-Barja, "Leveraging Mobile Edge Computing to Improve Vehicular Communications," 2020 *IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 2020, pp. 1-4, doi: 10.1109/CCNC46108.2020.9045698, Core ranking: B.

12. **N. Slamnik-Kriještorac** and J. M. Marquez-Barja, "Unraveling Edge-based in-vehicle infotainment using the Smart Highway testbed," 2021 *IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1-4, doi: 10.1109/CCNC49032.2021.9369622, Core ranking: B.

13. **N. Slamnik-Kriještorac**, M. Peeters, S. Latré and J. M. Marquez-Barja, "Analyzing the impact of VIM systems over the MEC management and orchestration in vehicular communications," 2020 29th *International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1-6, doi: 10.1109/ICCCN49398.2020.9209636, Core ranking: B.

14. **N. Slamnik-Kriještorac**, G. Landi, J. Brenes, A. Vulpe, G. Suciu, V. Carlan, K. Trichias, I. Kotinas, E. Municio, A. Ropodi, and J. M. Marquez-Barja, "Network Applications (NetApps) as a 5G booster for Transport & Logistics (T&L) Services: The VITAL-5G approach," 2022 *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 279-284, doi: 10.1109/EuCNC/6GSummit54941.2022.9815830, Core ranking: NA.

15. **N. Slamnik-Kriještorac**, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "An ML-driven framework for edge orchestration in a vehicular NFV MANO environment *Accepted*," 2022 *IEEE 20th Annual Consumer Communications & Networking Conference (CCNC)*, 2023, Core ranking: B.

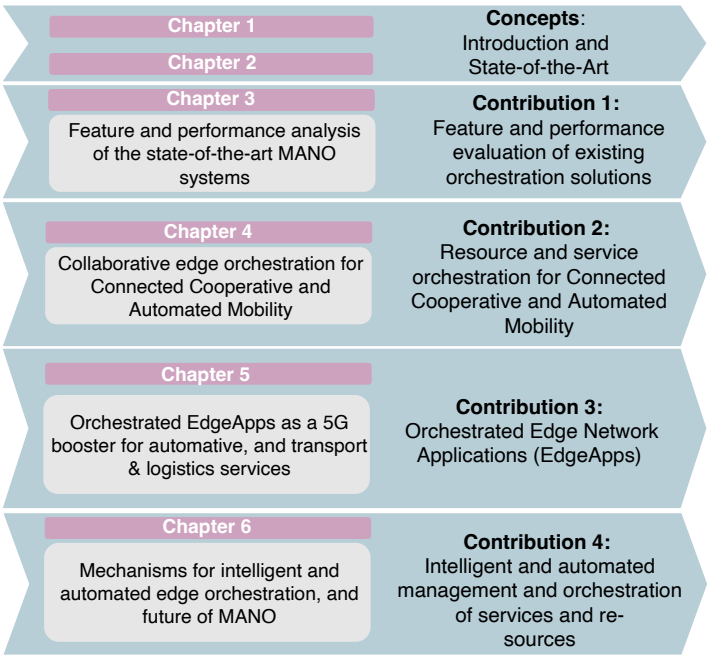


Figure 1.3: Mapping contributions to chapters.

16. **N. Slamnik-Kriještorac**, J. M. Marquez-Barja, "Mobile edge computing in Internet of Unmanned Things (IoUT) *(Submitted)*," in Emerging Technologies for Internet of Unmanned Things (IoUT) and Mission-based Networking, Springer.

1.4 Outline

Throughout this PhD thesis, the applicability of our research to the next-generation real-world networks and systems remained one of the main drivers of our day-to-day work. As such, our research has been enriched by an active and fruitful collaboration with several important industrial partners from 5G, automotive, and T&L sectors. In particular, the thesis contributed to the following European projects: H2020 5G-CARMEN, H2020 DAEMON, H2020 VITAL-5G, 5G-BLUEPRINT, and H2020 FED4FIRE+, resulting in five journal papers (published and submitted), and 12 publications in conference proceedings (full, work-in-progress, and demo papers; published and submitted). All the aforementioned publications are written as a first author, and listed in Section 1.3.

As shown in Fig. 1.3, this thesis consists of the six main chapters, which present the four main contributions. We start with the concepts in Chapters 1 and 2, introducing the research and outcomes, and presenting the state-of-the-art. In Chapter 3, we present the Contribution 1, elaborating on the Feature and performance analysis of the state-of-the-art MANO systems. Furthermore, we dedicate the entire Chapter 4 to Collaborative edge orchestration for Connected Cooperative and Automated Mobility (Contribution 2),

while Contribution 3, i.e., the Orchestrated Edge Network Applications (EdgeApps) as a 5G booster for automative, and transport & logistics services, is presented in Chapter 5. We close the thesis with the Contribution 4, which is covered by Chapter 6, through presenting Mechanisms for intelligent and automated edge orchestration and future of MANO systems. In the Appendix of this thesis, we present our survey on the Resource sharing in end-to-end 5G networks (Appendix A).

Chapter 2

State-of-the-art

In this chapter, we provide an overview of the state-of-the-art approaches that are relevant for each of the contributions presented in Chapter 1. The content of this Chapter is entirely based on the study of the related work that we conducted throughout the course of the work on the PhD thesis, and it helps the reader to understand the limitations in the existing approaches that motivated us to conduct research on this thesis.

We start this overview with the efforts on incorporating 5G and MEC to vehicular communications, thus, Section 2.1 presents the existing work on leveraging MEC in V2X context, as well as the insights into the management and orchestration solutions that could be used for orchestrating V2X resources and services.

In Section 2.2, we thoroughly investigate the existing management and orchestration solutions first, and then provide an overview of today's deployments for connected vehicles, thereby pointing out how the orchestration solutions can support CCAM service deployments.

To present the limitations in ongoing efforts on integrating network applications into 5G, Section 2.3 discusses the progress on defining the concept of EdgeApps, and elaborates further on the existing approaches for improving back-situation awareness on the roads. Finally, in Section 2.4 we introduce the efforts on making service orchestration intelligent.

2.1 Service and Resource Orchestration of MEC-enhanced Vehicular networks

In this section we provide an overview of works that motivated the incorporation of 5G and MEC to vehicular communications, aiming to achieve ultra-low latency as an ultimate goal. In Section 2.1.1 we overview existing works on leveraging MEC in vehicular context, while in Section 2.1.2 we present similar approaches to solve the insufficient flexibility and scalability of management and orchestration systems in MEC-enhanced vehicular networks. There is a number of works tackling MEC-based vehicular networks that are recently published, and within this section we present the ones which we consider important for the research direction of our approach.

Table 2.1: 5G and MEC in vehicular context.

Research direction		Work	
5G in vehicular communications		[1, 12]	
MEC in vehicular communications	MEC and Non-Orthogonal Multiple Access (NOMA)	[2]	[1, 12, 13]
	Computation offloading to MEC platforms	[14-17]	

2.1.1 Multi Access Edge Computing (MEC) in vehicular context

While studying the concepts of MEC and its benefits for vehicular environments, we created a brisk overview of works that study the incorporation of MEC in vehicular communications, as presented in Table 2.1.

In their comprehensive survey, Spinelli and Mancuso [22] explore how MEC is used in the context of vertical industries. In particular, based on their thorough study of literature, Spinelli and Mancuso provide some important conclusions on leveraging MEC in automotive sector, claiming that leveraging on the MEC host service deployments reduces latency up to 80% compared to existing network architectures without MEC deployments. Further improvements, especially in highly dense scenarios, could be achieved by combining MEC with several access technologies, such as 5G with sub-6 GHz, and mmWave deployments, which are enabling lower end-to-end latencies and higher throughput, and IEEE 802.11p, which is leveraging on unlicensed spectrum to provide wireless network access to vehicles that are consuming vehicular services deployed at the MEC hosts.

The involvement of cellular technologies to extend the awareness on the roads comes as a solution to the limitations imposed by Dedicated Short Range Communication (DSRC) based on IEEE 802.11p technology [23]. 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 [23, 24]. On the other hand, cellular technologies are characterized by larger coverage range, high network capacity, and technological maturity [23, 25]. Despite the aforementioned benefits, the centralized control in cellular networks causes additional delay against the strict delay requirements of safety vehicular applications [23, 26]. Thus, it is extremely important to carefully design the vehicular system, and to consider what technologies are suitable for a specific use case.

Since MEC relies on NFV to virtualize services, it is currently seen as a key platform for hosting diverse services, which can be discovered, accessed, and used, by vehicles [24]. In order to test a practical implementation of service provisioning in vehicular networks supported by edge and cloud, Laaroussi et al. [27] created an empirical analysis, by comparing the edge-based service provisioning, and the one provided by centralized cloud. Their results show that edge-based service provisioning outperforms the implementation with a centralized cloud in terms of achieved throughput, for the cases of different widespread application layer protocols, such as HyperText Transfer Procotol (HTTP), Constrained Application Protocol (CoAP), and Message Queueing Telemetry Transport (MQTT).

Furthermore, Ning et al. [28] agree that despite the benefits brought by MEC (e.g., highly efficient usage of mobile backhaul networks) there is still a vast room for improvement on ubiquitous connectivity, energy-efficient computation, and ultra-low latency [28]. As it is

Table 2.2: Management and orchestration of resources and services within MEC.

Research direction	Approach	Processes		Work
Closed-loop life-cycle management and orchestration	Theoretical	Orchestration	Challenges (vehicles' high speed, service continuity, high heterogeneity)	[6, 9, 8, 11]
			Programmable software framework for management and orchestration (SDN, NFV, MEC)	[6, 7, 8, 11, 18, 19]
			Overview of research projects	[9, 11]
		Control		[25]
		Monitoring		[20, 27, 28]
	Practical	Orchestration	MEC for 5G connected cars	[21]
		Control	MANO evaluation	[22]
		Monitoring		

estimated that more and more data will be processed by edge servers instead of centralized clouds due to their closer proximity from the end users [28], Ning et al. address the problem of offloading traffic from resource-constrained vehicles to MEC platform. They consider heterogeneous requirements of vehicle mobility and the computation tasks, integrating MEC-enhanced vehicular networks with Non-Orthogonal Multiple Access (NOMA) technology, which uses more efficiently the wireless spectrum [28].

As computation offloading is one of the benefits of bringing MEC to vehicular communications, there are numerous works that propose offloading schemes that optimize offloading decisions and resource allocation. However, task offloading supported by MEC is out of scope of our work in Chapter 3, and therefore more information can be found in [29, 30, 31, 32].

According to the survey provided by Spinelli and Mancuso [22], so far MEC has not been fully evaluated for vertical industries, as most of the literature presents MEC-based V2X frameworks and their architectures, with only few of them analyzing real datasets or evaluating performance of vehicular services in such a context. However, given the increase in realistic 5G and MEC testbed deployments, as well as the recent roll-out of 5G systems, this trend is expected to change in the upcoming years. In this thesis, and in particular from Chapter 3 onwards, we demonstrate the use of realistic experimentation environment based on the Smart Highway[1] and CityLab[2] testbeds, to test and validate the performance of MEC and NFV-based service deployments and their orchestration.

2.1.2 Management and orchestration of resources and services within MEC

In Table 2.2, we group background works into different categories based on whether they study orchestration, control, or monitoring, in a theoretical, or in a practical way. Regarding the MEC platform and its emerging features, Taleb et al. [33] present an extensive survey, which analyzes MEC as a decentralized cloud architecture that transforms the legacy Base

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

[2]CityLab: `https://doc.lab.cityofthings.eu/wiki/Main_Page`

Stations (BSs), into an IT environment at the edge of RAN. However, the orchestration of services and resources on a deployed MEC is recognized as a highly challenging task, due to the high speed of vehicles, and the need to maintain service continuity [33].

Furthermore, Soua et al. [34] discuss MEC in the context of vehicular communications, in order to meet the requirements of responsiveness, reliability, and resiliency for vehicular automated services. Within the literature scope that they spanned, Soua et al. [34] point at possible solutions for mobility-aware computation offloading, but they also focus on the resource management and orchestration challenges, mostly imposed by heterogeneity of resources and services at network edge. Soua et al. [34] and Taleb et al. [33] also discuss that this high heterogeneity in services, resources, technologies, and cloud infrastructure induce severe challenges to meet QoS and QoE requirements, and to maintain service continuity. Also, this heterogeneous nature makes the resource allocation [35] and management even more complex. Therefore, there is a need for more sophisticated framework for service and resource management, unifying networking and cloud orchestration [34, 33]. Importantly, Soua et al. [34] accentuate the need to exploit the synergy between NFV, SDN, and MEC, to create a programmable, flexible, and controllable architecture, particularly customized for Cooperative, Connected, and Automated Mobility (CCAM) use cases. Such architecture leverages on deployment of SDN controllers and Virtual Network Functions (VNFs), which should consider traffic characteristics, wireless diversity, and mobility patterns. As presented by Abdelaziz et al. [36], such management of the traffic handling is possible by enabling standard interfaces of the control layer, which further makes application layer more flexible.

In particular, MEC can benefit from SDN and NFV because of the opportunity to facilitate management and orchestration, putting it into an software-based framework. As shown by Liu et al. [37], the control component of management in SDN-based vehicular network assisted by MEC, runs on the commodity operation system. Thus, the deployment, update, and administration can be implemented by a software procedure. As an example for such management and orchestration platform, Soenen et al. [38] thoroughly present their modular and programmable management and orchestration framework that can be tailored to a service, or a particular VNF. According to Soenen et al. [38], the aforementioned customization can be achieved by constructing the so-called function-specific managers, and service-specific managers. These managers should be described and configured within VNF descriptors (VNFDs), and Network Service Descriptors (NSDs), so they support MANO entities towards managing and orchestrating a specific service and its resources in a custom manner.

Regarding the closed-loop life-cycle management and orchestration, there are several works which study concepts of the three constituent components (i.e., orchestration, control, and monitoring), but only separately. Apart from specific vehicular-based perspective, de Sousa et al. [39] distinguish and present some key concepts of network service orchestration, and also provide an in-depth taxonomy of different orchestration approaches and solutions, paving the way for the realization of diverse orchestration application scenarios. From a theoretical point of view, to realize network service orchestration a Multi-Domain Orchestrator (MDO) needs to be employed, as it coordinates resources and services in multiple administrative domains, spanning various technologies [39].

Moreover, both approaches presented in [33, 39] provide a valuable overview of the research projects relevant for network service orchestration, altogether with a number of orchestration

options that emerged from industry and standardization. In particular, de Sousa et al. [39] review the existing solutions from a more specific perspective than the general one provided in [33]. In their architecture-oriented overview, de Sousa et al. [39] studied the solutions based on the orchestration architecture, whether it spans one or multiple domains, in a hierarchical, cascade, or distributed manner, providing resource, service, or life-cycle orchestration, and so on. Regarding monitoring, there is an effort to study incorporating monitoring into the ETSI NFV architecture towards 5G, provided by Celdran et al. [40]. Their focus is on monitoring the control and data planes separately.

An interesting and yet realistic demonstration of using MEC for 5G connected cars is presented by Zhdanenko et al. [41]. This demonstration setup comprised cars, data collectors, analytical entities, and MEC orchestrator, showcasing also the impact of MEC server selection on the latency. In particular, the role of the data collectors is to aggregate the data from vehicles, such as GPS position and estimated changes in position over time. Furthermore, the analytical entities coordinate activities of all vehicles in their network in order to avoid collisions by pre-empting the next positions of vehicular traffic. The MEC orchestrator selects one MEC server based on any of the following criteria: i) static cloud (no migration), ii) distance-based MEC, iii) load-based MEC, and iv) distance and load-based MEC [41]. The total delay as a KPI would depend on the decision that MEC orchestrator takes. However, in their demo-based paper, Zhdanenko et al. present only a high-level architecture of the aforementioned system, without getting into details about particular orchestration solutions.

One of the rare attempts to test MANO systems has been introduced by Peuster et al. [42] recently. The main focus of their work is the test platform prototype that they developed to emulate up to 1024 Points of Presence (PoPs) on a single physical machine, which needs to be managed and orchestrated. Within the confines of their testing prototype, Peuster et al. [42] presented the concept of emulation-based smoke testing, used for automated, large-scale testing of two versions of Open Source MANO (OSM), i.e., OSM Release Three and Release Four.

Finally, Yilma et al. [43] propose a benchmarking setup for two ETSI NFV MANO solutions: OSM and Open Network Automation Platform (ONAP). Our approach to evaluate different MANO tools extends the perspective presented in the works we studied, since we: i) map the architecture of MANO solutions to the proposed closed-loop life-cycle management and orchestration, emphasizing its importance towards automation of network service and resource management and orchestration in vehicular communications, ii) provide an extensive analysis of the most utilized orchestration tools based on their features, and finally iii) compare two widely recognized MANO solutions, i.e., Open Baton (Release Six) and OSM (Release Six), based on their performance in terms of instantiation delay, and their isolated features. The thorough feature and performance analysis that we performed tackles the suitability of these existing MANO solutions for orchestrating realistic latency-sensitive vehicular applications, and their readiness to respond to dynamics in vehicular environment.

2.1.3 Virtualized Infrastructure Manager (VIM) systems in resource orchestration solutions

In order to design and develop real-time network services capable to cope with stringent QoS and Quality of Experience (QoE) requirements, containers are usually deployed as an alternative to Virtual Machines (VMs), as they usually demand low resource overhead, which makes them suitable for deployments on the resource-constrained network edge.

In their comparison between traditional VMs and containers, Doan *et. al.* [44] show that containers outperform VMs in terms of their suitability for MEC implementation, referring to specific service migration scenarios. However, the results provided by Salah *et. al.* [45] show that, although both deployed on top of the AWS Elastic Compute Cloud (AWS EC2), VM-based services outperform the container-based ones. This proves that impact of Virtual Infrastructure Manager (VIM) environment is not negligible, and it has to be studied deeper. Therefore, we see the potential of inspecting the type of VIM when approaching MANO of network services in highly dynamic and resource-constrained platforms such as MEC.

Furthermore, there are some efforts to evaluate the overall performance of existing MANO systems, but without focusing on the impact of specific MANO element in the ETSI NFV MEC architecture. For instance, the approaches presented in [33] provide a valuable but yet only theoretical overview of the orchestration solutions. In the previous section, we mentioned the attempt to evaluate existing MANO platforms by Peuster *et. al.* [42], emulating the Points of Presence (PoPs) that need to be orchestrated. Although the authors used two different version of OSM, there is no comparison of this tool to the other tools, and no discussion on how VIM influences the performance of OSM is provided.

Recently, there have been some efforts to benchmark different VIM environments, based on the self-generated performance reports. In their approach to measure the performance and VM instantiation times of OpenStack and Nomad, Ventre *et. al.* present the performance measurements of both VIM solutions, focusing on the tuning of performance for each of the VIMs. However, they do not focus on evaluating the impact each VIM has on the system, nor making the comparison between them. Sechkova *et. al.* [46] focus on the VIM, measuring the time overhead that VMs provisioning brings to the system. For this evaluation, Sechkova *et. al.* [46] conducted a comparative analysis of two open-source VIMs, i.e., OpenStack and OpenVIM.

To the best of our knowledge, our approach presented in Section 3 is the first which presents the evaluation of different VIM environments used as a NFV Infrastructure (NFVI) management system, thereby measuring the impact of VIM on the performance of particular MANO platform, within a real testbed environment.

2.2 Multi-domain orchestration of collaborative edges

In this section, we present an overview of related works on the collaborative orchestration in multi-edge 5G environments, focusing on the existing approaches first, and then discussing more on the ongoing efforts to make vehicles connected in distributed and orchestrated

network edges.

2.2.1 Existing approaches

As the focus of our work in Chapter 4 is on the management and orchestration of collaborative edges in the 5G ecosystem with distributed service deployments, this section provides an insight in the existing research efforts within related projects, reflecting on the features of existing NFV MANO solutions that need to be considered in order to properly design an orchestration platform.

According to the overviews provided by Taleb et al. [33] and de Sousa et al. [39], some of the open source orchestration tools that attracted significant attention in past few years are ONAP, OSM, Open Baton, Sonata (5GTango), Tacker, Cloudify, X-MANO, TeNoR, and Escape. The thorough analysis of NFV MANO solutions, which are either developed or utilized in most of the related projects, is presented in [47, 48, 49]. Such analysis is notably important for the development of our orchestration platform for CCAM, and associated network service and resource orchestration operations, because it provides a summarized information on the orchestration platforms, such as ONAP, OSM, Cloudify, among others, which are widely recognized in both industry and academia, serving as guidelines for future extensions of existing orchestrators.

Tackling *virtualization environment*, the cloud-native deployment of services in the orchestration platform for CCAM is following the principles of containerization, which enables deploying services and applications in a lightweight manner. Due to the MEC resource constraints, such lightweight deployment is particularly important for services running on the MEC platforms [47]. The support for containerization makes NFV MANO solutions (such as Open Baton, Sonata from 5GTango, latest version of OSM, Tacker, Cloudify, and Escape [47]), the valid candidates for orchestration and management of the latency constrained applications. Due to the support for various *monitoring tools* to be integrated in the orchestration platform for CCAM, and to provide orchestration entities with real-time information on the running edge services, it is possible to provide the VNF self-healing capabilities, decreasing the delay in communication between external monitoring tools and orchestration entities within platform. For example, a similar feature is available in ONAP, as well as Sonata from 5GTango.

Since ETSI is the leader in standardizing NFV and MEC, a corresponding NFV MANO tool should be designed and developed with reference to the ETSI NFV MANO framework. This in particular means that, although designed and developed by different vendors/operators/developers, different MEC platforms and applications can cooperate if they are following the standards. Therefore, our orchestration platform is carefully designed with respect to *ETSI NFV MEC framework* [50], aiming at extending current standards by defining reference points between mobile edge network orchestration functions.

The *multi-domain* capabilities represent a strong contributing factor to filter the orchestration solutions, being able to establish a connection with MEC platforms from the other edge domains using technologies such as OpenVPN and REpresentational State Transfer (REST), and to enable communication among different orchestration entities in multiple

domains. The multi-domain capabilities are of particular importance for our orchestration platform, as it provides distributed service deployment and collaboration between edges in 5G ecosystem. For instance, X-MANO solution [51] introduces the federation over multiple domains through the following core components: 1. Federation Agent (FA), associated to a particular domain in which it interacts with the domain orchestrators, and other modules which are in charge of the life-cycle management within a domain, 2. Federation Manager (FM), which is interfaced with one or more FAs, and 3. OpenVPN as a cross-domain link. Another solution that also considers the federation aspects is ONAP, as its modular and layered nature improves interoperability and simplifies integration, allowing it to support multiple VNF environments by integrating with multiple VIMs, VNF Managers (VNFMs), SDN Controllers, etc. In particular, ONAP's service orchestrator performs orchestration at a high level, with an end-to-end view of the infrastructure, network, and applications. Moreover, ONAP's multi-site state coordination module enables scaling to multi-site environments to support global scale infrastructure requirements. Certain process specifications and policies are geographically distributed to optimize performance and maximize autonomous behavior in federated cloud environments. Furthermore, Escapev2 Orchestrator [52] provides multi-domain NFV orchestration by: i) performing recursive orchestration via north and south Unify interfaces, supporting different legacy technologies and migration between them, and ii) supporting Unify domains directly, and several technological domains via adapters. Finally, TeNoR [53] defines VNF orchestration as a multi-domain problem, considering several Points of Presence (PoPs) in the NFV infrastructure. The TeNoR orchestrator, as a product of FP7 T-NOVA project, is responsible for network services and VNFs' lifecycle management operations over distributed and virtualized network/IT infrastructures.

Although the aforementioned management and orchestration solutions are mature and robust, tackling an end-to-end perspective in virtualized network infrastructure, they are still lacking the support for automated edge-to-edge service deployment that anticipates highly challenging mobile scenarios, thereby enabling fast orchestration operations across different network edges. Another missing link is coupling with 3rd Generation Partnership Project (3GPP) systems, such as 5G, and design of platform and its operations in accordance with the overall 5G ecosystem. Thus, to enable service continuity in such challenging 5G ecosystem, the orchestration platform that we present in the Chapter 4 responds to the aforementioned challenges by enabling collaboration between i) orchestrated network edges themselves, and ii) between edges and the 5G System, while taking into account high mobility, and resource and service demand. In such platform, all orchestration tiers collaborate in their orchestration operations for intrinsically distributed service deployments, via fast and dynamic set-up of the management and orchestration reference points between mobile edge network orchestration functions, and by providing an automated orchestration at and between edge networks within the same or different administrative domains (i.e., MNO's domain, country, etc.).

2.2.2 Connected vehicles in distributed network edge environments

To extract the potential of providing the localized access to virtualized network resources and services in the 5G ecosystem, i.e., the ultra-reliable and low-latency service deployments, challenges such as resource constraints in network edges, and high user mobility,

Table 2.3: Overview of today's deployments for connected vehicles.

<table>
<tr><th>Technology</th><th>DSRC</th><th colspan="4">Cellular</th></tr>
<tr><td>Benefits</td><td>scalability</td><td colspan="4">large range (i.e., service coverage increased), high capacity, and technological maturity</td></tr>
<tr><td>Challenges</td><td>narrow service coverage, increased communication load, inefficient congestion control, insufficient reliability</td><td colspan="4">additional delay due to the centralized control</td></tr>
<tr><td colspan="2" rowspan="2">Type</td><td colspan="2">LTE V2X PC5 sidelink/NR V2X PC5 sidelink</td><td colspan="2" rowspan="2">V2N - Uu based/
+ support from collaborative orchestration</td></tr>
<tr><td>mode 3/mode 1</td><td>mode 4/mode 2</td></tr>
<tr><td colspan="2">Characteristics</td><td>radio resources allocated via cellular network</td><td>radio resources allocated simultaneously</td><td colspan="2">vehicles allocating radio resources via cellular network, communication performed via Uu interface</td></tr>
<tr><td colspan="2">Suitable use cases</td><td colspan="2">road context information sharing in a close proximity</td><td colspan="2">safety, non-safety, and infotainment V2X use cases that span multiple edge domains</td></tr>
<tr><td colspan="2" rowspan="4">Challenges</td><td colspan="2" rowspan="2">service coverage includes strongly limited number of vehicles</td><td>V2N-Uu based only</td><td>V2N-Uu + orchestration</td></tr>
<tr><td>dynamic provision of V2X services due to the high mobility</td><td>multi-edge service deployment</td></tr>
<tr><td colspan="2">maintaining connectivity between vehicles due to the high mobility</td><td>maintaining service continuity</td><td>edge-to-edge service continuity</td></tr>
<tr><td colspan="2">burden for computing capabilities of a single vehicle due to the broadcast mode of CV2X messages</td><td rowspan="2">achieving application portability and immutability</td><td rowspan="2">cloud-native service deployment</td></tr>
<tr><td colspan="2">insufficient information for network (re)selection</td></tr>
</table>

need to be properly addressed. These challenges become even more severe when considered in highly mobile environments with connected vehicles, since they require continuous monitoring of network and computing resources, fast reconfiguration of service deployments in distributed edges, and following the user mobility patterns, as well as their associated service and resource demand, which all fall under the umbrella of *network resource and service management and orchestration* tasks.

Since connected vehicles are a valid representative of highly mobile users, in this Section we focus on the automotive class of use cases, as a 5G ecosystem vertical, and investigate the challenges that need to be properly tackled by collaborative service management and orchestration platforms to enable service continuity. As illustrated in Fig. 2.1, virtualized network services are deployed on top of the distributed edge clouds, and there are different communication technologies that enable connectivity between vehicles and services, and between services themselves. Thus, in Table 2.3 we provide an overview of these network technologies, focusing on the benefits of cellular networks and their coupling with orchestrated edge service deployments, and identifying the bottlenecks that can be mitigated by collaborative orchestration (as shown in the top right column of the Table 2.3).

To alleviate issues on the roads imposed by insufficient cooperation between vehicles, a significant effort is being invested by automotive industry, MNOs, and research institutions, toward enabling vehicles and surrounding infrastructure with the communication capabilities. If equipped with communication engines, vehicles can share information about different events not only with surrounding vehicles (i.e., via DSRC and PC5 sidelink), but also with those in a larger vicinity, thanks to the cellular networks and distributed service deployments (Fig. 2.1).

As presented in Table 2.3, the DSRC based on IEEE 802.11p technology, as well as cellular

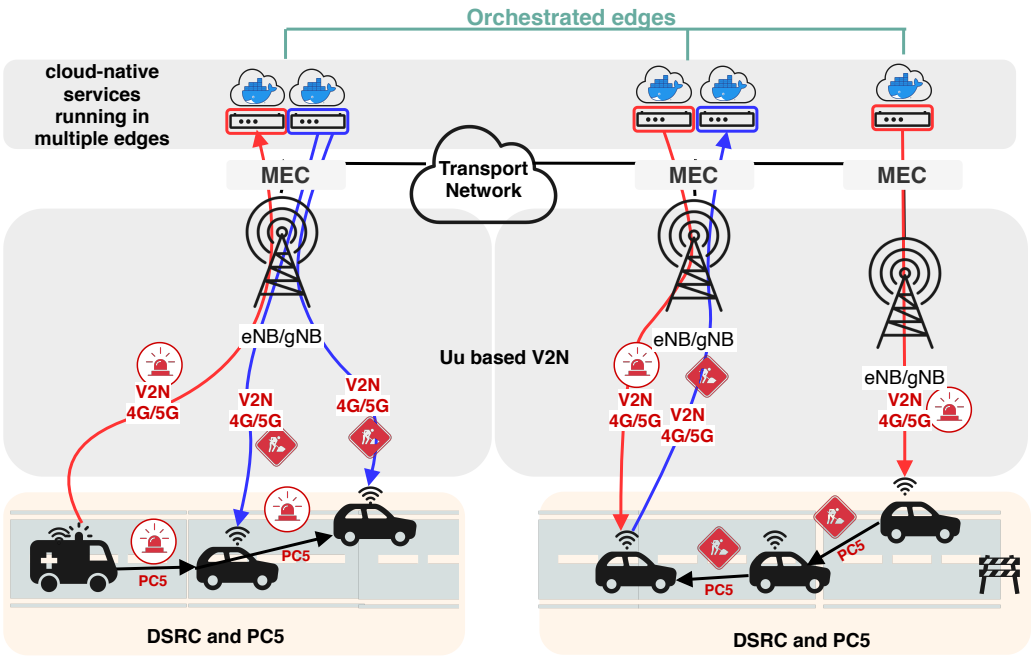


Figure 2.1: 5G V2X vehicular communications supported by collaborative orchestration.

PC5-based communication, impose constraints to realistic V2X scenarios in comparison to Vehicle-to-Network (V2N) Uu-based communication. Due to the short range that is covered by internet gateways in case of DSRC [54], and communication only with the vehicles in close proximity in case of PC5 [54, 55, 56, 57], it is challenging to cope with high mobility of vehicles on the highways, and to handle the use cases that require extended awareness that spans multiple administrative domains (e.g., countries). Such gaps can be efficiently bridged by utilizing cellular network infrastructure [54, 58, 59, 60].

The cellular infrastructure provides sufficient information for: i) central controllers to efficiently decide on the handover timing [54], and ii) service orchestrators to perform proactive service deployment and service migration from one edge to another. This is not possible in the case of DSRC and PC5 where the local information that each vehicle contains does not involve a broad view of the overall network, thereby leading to inefficient network (re)selection, and reactive service instantiation and migration, which lead to disruptions in service performance. Despite the improved KPIs promised by 5G (i.e., ultra-low latency, high bandwidth, etc.), if management and orchestration of resources and services are not present in the cellular systems, service continuity will not be ensured due to the lack of collaboration between network edges. Thus, Fig. 2.1 illustrates the multi-edge deployment of cloud-native services that can be efficiently migrated from one edge to another, as a result of management and orchestration operations that take into account the resource constraints, as well as high user mobility. In the second column of Table 2.3, we emphasize how collaborative orchestration can further improve V2N - Uu based communication and to support connected vehicles by mitigating the challenges of i) dynamic provision of V2X services due to high mobility by performing multi-edge service deployment (as described in Section 4.1.3), ii) maintaining service continuity by enabling edge-to-edge service continuity (as described in

Section 4.1.3), and iii) achieving application portability and immutability by applying cloud-native service deployment (as described in Section 4.1.4), which further facilitates service relocation. The aforementioned benefits for 5G V2X systems are the outcome of collaborative orchestration, thus, it needs to be efficient and robust, as any delay or interruption in performing an orchestration task (e.g., service instantiation, scaling, and termination) can significantly impact the deployment and operation of services used by vehicles, thereby leading to e.g., uncoordinated maneuver recommendations, or outdated instructions. Thus, it is essential to carefully study the orchestration concepts, and to build efficient orchestration solutions, to be able to make use of multi-edge deployments and edge-to-edge relocation of cloud-native services, performed in a timely manner.

2.3 Orchestrated Edge Network Applications for 5G Vertical Services

With reference to our work in Chapter 5, this section provides insights into state-of-the-art work on the orchestrated network applications for 5G verticals. First, we discuss the current position of EdgeApps in 5G ecosystems, and then we detail more on the works related to extending back-situation awareness in the automotive vertical, by studying the related to concepts to our 5G BSA application service presented in Section 5.3.

2.3.1 Edge Network Applications for 5G and beyond verticals

The proliferation of 5G deployments will undoubtedly spawn new opportunities for numerous vertical industries, including manufacturing, automotive sector, e-health, and transport & logistics. In [61], Malandrino and Chiasserini study the potential of different industries, i.e., the high-traffic applications, to become 5G verticals and gain from integrating 5G in their day-to-day operations. They performed such an analysis based on a large-scale, real-world, crowdsourced mobile traffic trace, and they also made a classification of the existing applications based on their total traffic, peak rate, and sparseness [61]. The outcome of their analysis reflects on the large group of applications that could actually benefit from 5G integration, where most of them belong to major over-the-top content providers, while further at a more general level, Malandrino and Chiasserini [61] derive an important justification of leveraging 5G in all emerging applications.

In their work on the advanced 5G architectures for future EdgeApps and verticals [62], Patachia et al. provide a telco-oriented perspective on the deployment of EdgeApps, focusing on the adjustments that need to be accommodated in the 5G network itself. They identify the gaps in current network deployments of telco operators, which hinder the implementation of innovative use cases, and then propose the adaptations such as DevOps and AI/ML-based cognition, which need to be deeply integrated in the telco network infrastructure to enable end-to-end network automation capabilities. Such adaptations will be applied through several future 5G functionalities and services, i.e., i) EdgeApps on-boarding, which enables managing EdgeApp packages from various tenants, ii) EdgeApp experimentation APIs that will expose standardized OpenAPIs to provide access to the lifecycle management

of EdgeApps and EdgeApp catalogues, iii) EdgeApp orchestrator, which will be in charge of the overall EdgeApp deployment, iv) MANO client API (SOL005) service that interfaces experimentation and operation with EdgeApp orchestrator, and v) Continuous Integration and Continuous Delivery (CI/CD) service that will provide CI/CD pipelines to coordinate the execution of tests by interacting with various orchestrators. In line with that, Bonea et al. [63] present their recent work and progress on building the framework for testing and validation of EdgeApp, which is based on a completely equipped 5G testbed with connectivity to 5G Non-Standalone (NSA), and ONAP selected as an orchestrator of EdgeApps.

Furthermore, Patachia et al. [62] envision that the changes applied in 5G networks will pave the way towards an increased development and testing of 5G EdgeApps, thereby enabling dynamic allocation of 5G network, computing and storage resources, as well as flexible deployment of vertical services in distributed cloud infrastructures. However, the aforementioned EdgeApp-oriented 5G frameworks are not standardized yet, and as of early 2021, there are several European projects that focus on EdgeApps and their design and development, which progress in research directions that will support vertical industries towards better understanding and integration of 5G in their service paradigms.

In particular, based on the overview of satellite network integration in the 5G ecosystem studied and experimented in the 5GENESIS project [64], Fornes-Leal et al. [65] demonstrate how an integration of satellite backhauling can extend 5G coverage to the rural and underserved areas by deploying 5G applications on the network edge, as a part of a smart farming use case. The concept of EdgeApps that we propose and present in the Chapter 5 could be also leveraged in such use cases, where the requirements on the bandwidth and low latency to enable faster field sweeps, higher accuracy, and lower energy consumption, can be also embedded in the EdgeApp blueprints and descriptors, as further described in Chapter 5. Another initial work on the EdgeApps is given by Apostolakis et al. [66] related to designing EdgeApps tailored for Public Protection and Disaster Relief (PPDR) use cases, which will be deployed in a fully virtualized containerized 5G network within the 5G-EPICENTRE project. For the PPDR use cases the benefits of such work will be two-fold: enhancements in the network performance, and automated operations supported by Kubernetes (K8s)-based support.

Finally, in [67], Trichias et al. presented a comprehensive overview of the Vital-5G project, thereby spanning the Vital-5G platform, the three trial sites and use cases (Antwerp sea port, Galati river port, and Athens warehouse/hub), as well as key innovation and commercialization aspects. On the other hand, the focus of our work in Chapter 5 is particularly on the EdgeApps, detailing on the EdgeApp structure and packaging, their unique role in enabling T&L services to leverage 5G capabilities, followed by a few examples of real-life EdgeApps that are designed for improving the safety and efficiency of operations in the river port.

Before we close this section, let us give an overview of the so-called Service Enabler Architecture Layer (SEAL) architecture, which is standardized by 3GPP [1] as a part of Release 16, and an effort to address an ever-increasing demand for vertical applications. As an effort to enable operation of such applications in 5G and beyond systems and to cope with the proliferation of vertical industries, 3GPP is fostering an innovation in the application layer, focusing on standardization of vertical applications [68]. In line with that, Shah et al. [68] provide an overview of the SEAL standard and its position in the 5G network, explaining how

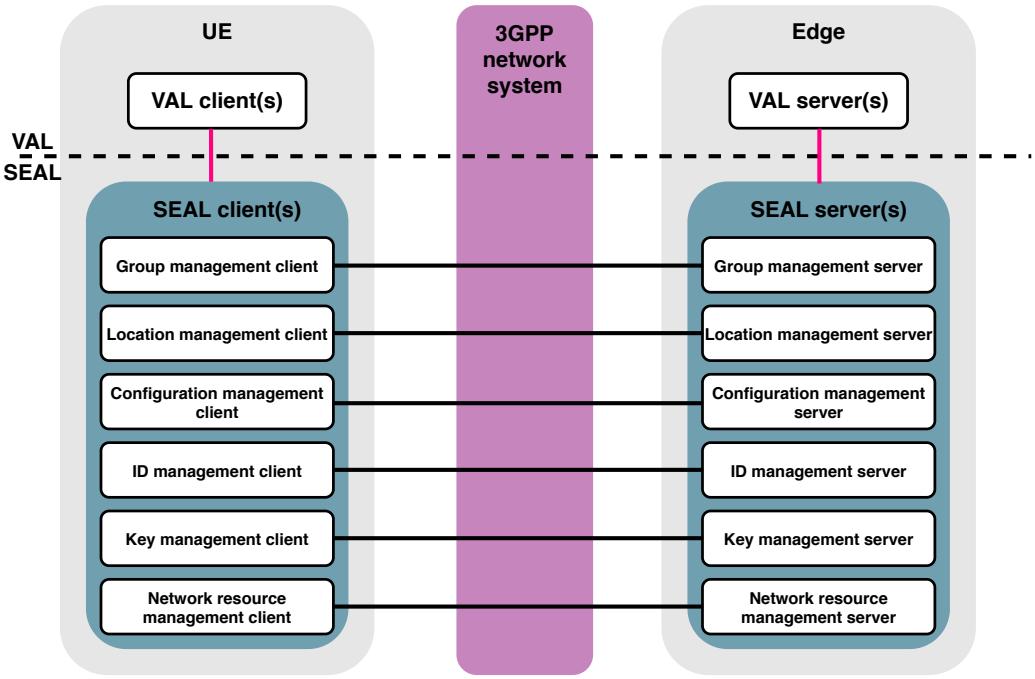


Figure 2.2: Service Enabler Architecture Layer (SEAL) [1].

vertical industries can leverage SEAL to efficiently develop and deploy their applications in the 5G ecosystem. According to Shah et al. [68], the fundamental goal of creating SEAL architecture is to facilitate the development of vertical applications by enabling developers to completely focus on the core functionalities of their applications, i.e., Vertical Application Layer (VAL), and further leverage SEAL for the auxiliary services that could help core ones integrate better with 5G systems.

The functional architecture of the SEAL framework is shown in Fig. 2.2, and we list the main features below:

- *VAL Client*, which is responsible for providing client functionalities specific to vertical applications, and the same time interacting with the VAL server and SEAL clients (e.g., V2X UE client).
- *VAL Server*, which provides a server functionality specific to vertical applications, thereby interacting with VAL client and SEAL servers (e.g., V2X application server such as the BSA one that we present in Chapter 5, Section 5.3).
- *SEAL Client*, similarly as VAL, provides client side functionalities specific to SEAL service, and interacts with the VAL client and SEAL servers.
- *SEAL Server* is responsible for providing server side functionalities specific to SEAL service, while interacting with the SEAL clients and VAL servers.

As it could be noticed from the aforementioned description, VAL clients and VAL servers provide vertical application specific functionalities, while the SEAL clients and SEAL servers

provide a common framework as a support for multiple vertical applications. Making a parallel with our approach of EdgeApps described further in Chapter 5, VAL server corresponds to vertical-specific EdgeApps, while SEAL servers are in line with our definition of vertical-agnostic EdgeApps, which could be leveraged by different vertical services. Some of the most common SEAL services are: i) Location Management (LM) that provides sharing location data between client and server for vertical application usage, ii) Group Management (GM), which allows vertical applications to manage a group communication, i.e., to create and manage the group, as well as group specific policies and group members, iii) Configuration Management (CM) provides initial configuration for all users and notifies them in case any change in the configuration happens, while supporting creating and maintaining UE configuration and user profile configuration for vertical applications, iv) Identity Management (IM) supports users' authentication and authorization, v) Key Management (KM) supports secure generation and distribution of encryption keys to VAL users, and vi) Network Resource Management (NRM) enables vertical applications to switch and manage radio bearers (unicast and/or multicast).

However, even such thoroughly defined architecture of future vertical applications still does not take into account the QoS requirements specific for the vertical, which is a complex task given that vertical users (e.g., vehicles, automated guided vehicles, and vessels) connect to different services in the backend, either in the edge or in the cloud that belong to different beneficiaries. With regard to the QoS-awareness, Du et al. [69] propose a service-aware network architecture for vehicular systems with service-network mapping mechanism, which binds the specific service application to its traffic packets. In their study, Du et al. [69] consider that different applications are running on a vehicle, and that each of them requires different QoS requirements that are specified in the trailers of the traffic packets. Furthermore, they leverage on the UpLink CLassifier (ULCL), which is a functionality within User Plane Function (UPF) of the 5G Standalone (SA) core, configured with traffic rules to steer the traffic coming from the vehicle based on the trailers of the packets, which are treated as a part of the payload. As such approach inevitably adds an overhead to each packet that vehicles send out, this might increase the overall backhaul traffic produced by each vehicle. The rules need to be defined for each type of the application on the UPF, which might also hinder the 5G network operation given the ever-increasing number of applications. In our approach presented in Chapter 5, EdgeApps include 5G-related requirements in their blueprints, thus, even before the deployment phase, and then the deployment is tailored to specific 5G needs of the EdgeApp by assigning it corresponding network slices. Hence, there is no overhead to the traffic packets produced by EdgeApps and there are no other service-specific requirements that need to be handled by UPF and ULCL.

2.3.2 On Improving Back-Situation Awareness

In Chapter 5, we focus on a particular vehicular use case, i.e., back-situation awareness on the highways, and show how EdgeApps should be designed for such a vertical service. Thus, here we briefly present the existing work on enhancing awareness about emergency situations on the roads. By exploring the opportunities of technological advancements, the Emergency Management System (EMS) providers are making an effort to optimize the use of existing resources, and to offer high-quality medical services to the patients. Some of

the key goals of EMSs are reducing patient mortality, preventing disability, and improving chances of recovery [70]. 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 the emergency events. According to Sánchez-Mangas et al. [71], the probability of death decreases by one third, if there is a 10 min reduction in the emergency response time. Furthermore, Vukmir [72] 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 [71, 72].

Another study, provided by Iannoni et al. [73], shows how the extensions of the hypercube model, combined with the hybrid genetic algorithms, optimize the configuration and operation of Element Management System (EMS) on the highways. Considering the locations of the ambulance bases along the highway, Iannoni et al. [73] 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.

Nowadays there is a strong focus on the use of vehicular systems and the V2X applications that are developed to improve safety on the roads [24, 74], thereby enhancing the situation awareness [75]. For instance, Siegel [76] presented a system that is able i) to receive a message from an EmV, which carries the information on EmV's ID, position, speed, direction, route, etc., ii) to determine a route of the EmV, as well as the routes of other vehicles in vicinity of this EmV, and iii) to alert vehicles in vicinity about the EmV's presence. Another approach on this topic is also found in the work presented by Hadiwardoyo et al. [77], where they propose a Vehicle-to-Vehicle (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, in such an approach, information about arriving EmV is only shared with the vehicles in a 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 manoeuvre.

A study conducted by Senart et al. [78] 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 a 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 approach is that it increases the response time of an EmV.

The situation awareness is also studied by Metzner and Wickramarathne [79], where the V2V technology is used for sending the required notifications. However, as already mentioned in the context of work presented by Hadiwardoyo et al. [77], 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 [80], who propose using Vehicle-to-Infrastructure (V2I) communication, i.e., making use of 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 [24].

The efforts to extend the range of notifications by utilizing cellular technologies are presented by Shah et al. [24], as well as in our previous work [81], where we study the use of 5G systems and MEC to support vehicular use cases, thereby decreasing the delay in communication by deploying vehicular applications at the network edge.

In the research [82, 83, 84], 5G New Radio (5G NR) is recognized as an enabler of ultra-low latency and high reliability of the 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 [83]. However, Wang et al. [84] point out that the use of 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, Halili et al. [85] study the benefits of using 5G systems 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 the Chapter 5, Section 5.3, we present the edge-aware MEC application service that is developed for enhancing the back situation awareness on the highways, relying on the 5G technology to provide connectivity to vehicles. Furthermore, we present the operational aspects of such service in a multi-domain deployment, thereby enabling the extended awareness about the EmV along the road that spans several edge domains.

As emphasized already at the beginning of this section, the response time to the incidents is considered as the most essential performance indicator [86, 87]. Accordingly, several attempts are made to model, and to predict, the response time. One of these attempts is provided by Poulton et al. [88], who present the work on modeling metropolitan-area ambulance mobility under blue light conditions. This work uses historical data collected internally by the emergency ambulance services, but it does not consider the real-time information, traffic, nor the context-related information retrieved from the external traffic management systems.

Another approach is presented by Kapileswar et al. [89] where Frequency and Distance-based Priority MAC (FDP-MAC) protocol is used to broadcast warning messages to both pedestrians and vehicles, and they provide a simulation setup for such an information dissemination system within the NS-2 simulation environment. The experimental assessment of the BSA application service presented in the section 5.3 is conducted in a realistic testbed environment, in which we enable the reduction of the overall emergency response time.

2.4 Intelligent Edge Orchestration

Due to the plethora of computing resources, the remote cloud centers have been a prominent solution for offering means for hosting and maintaining various Internet-based applications [90]. However, recent technological trends (e.g., Industry 4.0, automotive sector, and transport & logistics) introduced new challenges that push an urgent need for changing computer and network architectures due to the significant increase in number of connected devices [91]. All these connected devices, such as smartphones, wearables, IoT devices,

vehicles, etc., used to send their information to cloud data centers for further processing and decision making [90, 91]. Since cloud resources are usually located at geographically distant computing machines, the increased delay in communication between the device where data originated, and the cloud, impose significant challenge for achieving required levels of QoS and QoE [90, 91, 47, 92]. Besides the increase in communication delay, there is also an increase in computation on the cloud (i.e., computational delay), due to the need to process an enormous amount of traffic coming from hundreds and thousands of devices [91]. Thus, the edge computing paradigm has emerged as a promising solution to improve network performance in 5G, while reducing computational delay, transmission delay and bandwidth consumption, via exposing computing resources at the network edge, i.e., closer to the end users/devices where data is produced/consumed [93]. However, opposed to resources in the cloud, edge resources are i) constrained, i.e., the amount of computing resources is limited due to the smaller processors and a limited power budget [90], ii) heterogeneous, as resources might belong to different vendors, and iii) dynamic, i.e., nature of edge resources is fluctuating due to the changes in workload, traffic demand, and users' mobility [90, 93]. Therefore, a proper management of such resources needs to be ensured, in order to use the computing and network resources in an optimized manner.

However, considering strict requirements for 5G networks (e.g., greater coverage, end-to-end latency of 1-10 ms, massive connectivity, and massive capacity), together with the enormous dynamicity and heterogeneity in edge resources, traditional manual network management becomes impossible to scale and to maintain [93, 47]. Hence, there is a need for transition from a poorly scalable network management to its automation, thereby providing a solution for sophisticated management and orchestration that is at the same time compliant to the standards, such as ETSI NFV, ETSI MEC, and 3GPP [47]. In this section, we provide an overview of state-of-the-art MANO solutions that are widely used in both research and industry, with an insight into some recent efforts to apply AI/ML to solving edge resource allocation and complex decision-making problems.

Edge orchestration can benefit from SDN and NFV because of the opportunity to design a MANO solution as a software-based framework, which can run on a commodity operation system with the procedures of deployment, update, and administration, implemented as a software procedure. One example of such approach is presented by Soenen et al. [38] who built a modular and programmable MANO framework tailored to a service, or a particular VNF. Such service-specific customization is enabled by so-called function-specific managers, and service-specific managers, which are described in VNFDs, and NSDs.

Some of the open source orchestration tools that attracted significant attention in past few years are ONAP, OSM, Open Baton, Sonata, Tacker, Cloudify, X-MANO, TeNoR, and Escape [33, 39] and the functional analysis of a subset of these MANO solutions is presented in [47, 48, 49]. To compare and study different orchestration solutions, there are different aspects to consider, as already presented in 2.1 and 2.2.

However, an agile operation with automated incorporation of changes in service deployments still remains challenging in most of the existing MANO solutions. For addressing such challenge, [94] identifies ML and in general AI as key enablers for increasing automation, where AI-powered mechanisms require a fast access to data, abstraction of intelligent and contextual information from events and rule-based systems, supervision, streamlined workflows and lifecycle management. With the support from AI/ML, network optimization can be per-

formed at different timescales, thereby enabling more intelligent MANO operations, which is currently not specified by ETSI NFV MANO. As stated in [94], ETSI ISG NFV considers incorporating AI/ML into their already standardized MANO stack, although the ETSI NFV is not explicitly considering AI/ML for applications in operation automation but rather in requirements to properly feed data and collect actions from AI/ML modules [95]. Currently, the automation mechanisms in NFV MANO are rule based auto-scaling and auto-healing provided as policies in VNF descriptors. These descriptors provide the description of the scale or heal actions to be executed when a condition involving monitoring parameters or VNF indicators is satisfied. The enforcement of auto-scale or auto-heal rules occurs in the VNF manager and the NFV Orchestrator (NFVO) automatically. However, automation procedures are not tackled by ETSI NFV MANO but by standardization groups such as ETSI ISG Zero-touch Network and Service Management (ZSM) [96, 97], and ETSI ISG Experiential Networked Intelligence (ENI) [98, 99].

Let us take a look at different classification techniques for handling resource management in cloud and edge computing.

- *Discovery* is used to find available resources from the cloud, fog, or edge layers, based on workload requirements, and to identify where the workload can be deployed efficiently. It is performed by using a manager or master entity that has an overall view of the resources (e.g., the role of the orchestrators). Afterwards, based on the workload requirements, this manager can allocate resources properly among fog and cloud layers. According to Hong and Varghese [90], in the edge/fog computing concept, the discovery algorithms stand for determining resources in the edge network that can be employed for further distributed processing, and based on analysis provided in [90, 92] some of the mostly utilized edge resource discovery algorithms are Round Robin (RR), Equally Spread Current Execution (ESCE), Shortest Job First (SJF), Gaussian Process Regression for Fog–Cloud Allocation (GPRFCA), and Remote Sync Differential Algorithm (RSDA).
- *Load balancing* distributes the workload to edge nodes to make the operations more efficient by avoiding congestion, low load, and overload. Accordingly, some of the prominent edge load balancing algorithms are Dynamic Resource Allocation Method (DRAM), Efficient Resource Allocation (ERA), Priority based Resource Allocation (PBSA), Feedback-Based Optimized Fuzzy Scheduling Algorithm (FOFSA), Hill Climbing Algorithm (HCA), Efficient Load Balancing Algorithm (ELBA), and Tabu Search Algorithm (TSA) [92].
- *Placement* is used to determine the suitable resources to satisfy the required workload. The main purpose is to distribute the incoming computation tasks to the appropriate fog/edge resources. The iterative algorithm based on resource placement is a method that performs three algorithms: i) it first sorts the edge nodes and application modules according to their environments, ii) it looks for an eligible edge mode that meets requirements, and iii) it is responsible for ensuring the requirement check.

As one of the attempts to optimize the resource management techniques, presented earlier in this section, Fu et al. [93] propose an AI-assisted intelligent wireless network architecture, and based on the proposed architecture, they propose a Deep Q-network (DQN) algorithm to

figure out the complex and high-dimensional joint resource allocation problem. Their simulation results show that the algorithm has good convergence characteristics, as the proposed architecture and the joint resource allocation scheme achieve better performance compared to other resource allocation schemes. In their approach, the communication, computing and caching resources are virtualized and provided in resource pools, which orchestrator jointly manages and orchestrates by applying AI algorithms. The orchestrator, with the built-in AI algorithm, analyzes the system resources status and task attributes to dynamically allocate corresponding computing, caching and communication resources for specific tasks. Although it sounds promising, this DQN-based solution poses significant challenges in edge computing environments, due to excessive amounts of resources it requires for training and inference. Although suitable for high-dimensional complex resource allocation problems, some further issues might imply from applying DQN. For example, due to an enormous scale of the network, it is difficult to perform the training process in an online manner, as it consumes a lot of time and computing resources. Thus, DQN-based schemes need to be trained offline, while further adjustments can be applied online when needed [93]. Such adjustments might not be efficiently decided and applied in case of extremely time-sensitive tasks, which ultimately leads to inefficient neural networks causing bad actions that result in failed tasks. Furthermore, dimensioning of neural networks is an extremely difficult tasks in the wireless network environments, given the ever-changing topologies, where the number of working edge computing nodes, access points, caching servers, and UEs, fluctuate all the time. Any change in the topology would mean that an already trained neural network needs to be trained again. Finally, the main obstacles for creating and applying most of the AI/ML models are the dimensionality of learning and the complexity of decision making [93]. To improve the decision-making, large amounts of data are required for the training phase. As such training requires computational capabilities that are usually not present in the edge environments, Fu et al. [93] propose using distributed ML, which will include data parallelism, i.e., distribute the same model to different computing machines that will separately train the model using smaller datasets. Hence, a final product will be to merge all outputs from the distributed training engines.

As distributed edge environments are complex, the application of AI/ML may not be a straightforward task given the diversity of orchestration operation that need to be optimized. Thus, to study this issue with a more prominent attention, in Chapter 6, we conducted a thorough analysis of different AI/ML techniques, and their suitability for edge environments and service orchestration. In addition, we have also presented our attempts on applying AI/ML techniques on orchestration operations, showcasing the results that reflect on the potential for improving decision-making by making proactive service/EdgeApp deployments and relocations from one edge to another.

Chapter 3

Feature and performance analysis of the state-of-the-art MANagement and Orchestration (MANO) systems

This chapter is part of the **Contribution 1**: *Feature and performance evaluation of existing orchestration solutions*, and it is based on:

N. Slamnik-Kriještorac, E. de Britto e Silva, E. Municio; H.C. Carvalho de Resende, S.A. Hadiwardoyo, J.M. Marquez-Barja, "Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context," in *Sensors* 2020, 20, 3852. https://doi.org/10.3390/s20143852

N. Slamnik-Kriještorac, H. C. Carvalho de Resende, C. Donato, S. Latré, R. Riggio and J. Marquez-Barja, "Leveraging Mobile Edge Computing to Improve Vehicular Communications," 2020 *IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 2020, pp. 1-4, https://doi.org/10.1109/CCNC46108.2020.9045698

N. Slamnik-Kriještorac, M. Peeters, S. Latré and J. M. Marquez-Barja, "Analyzing the impact of VIM systems over the MEC management and orchestration in vehicular communications," 2020 29th *International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1-6, https://doi.org/10.1109/ICCCN49398.2020.9209636

N. Slamnik-Kriještorac and J. M. Marquez-Barja, "Demo Abstract: Assessing MANO Performance based on VIM Platforms within MEC Context," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 1338-1339, doi: https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162932

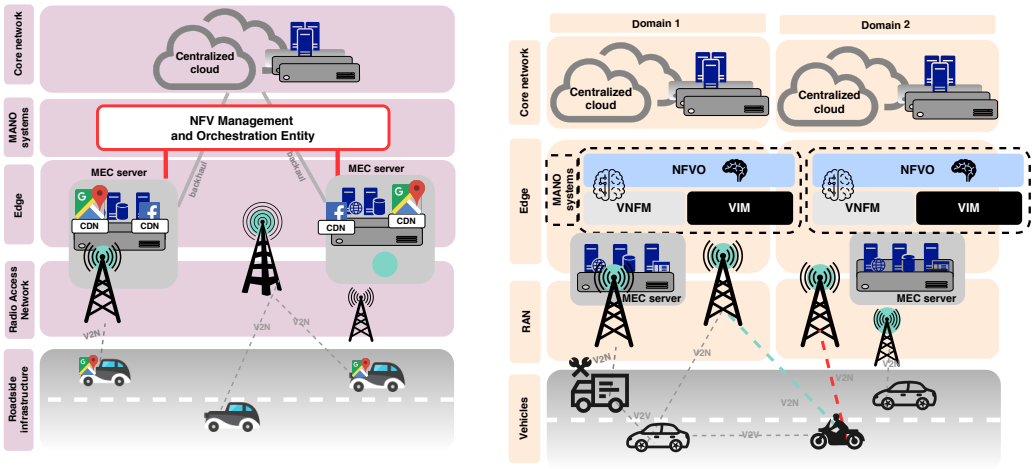
The fluctuating nature of available resources and traffic, considered jointly with highly stringent requirements for network services, imposes rigorous conditions for operation in communication networks nowadays. It is of paramount importance to proactively address upcoming network service requests, and yet an appropriate network management regime is necessary

in order to satisfy users' requirements and expectations. However, the aforementioned constraints are even more emphasized in case of vehicular communication networks, where each vehicle becomes a frequent participant in communication [24]. Such collection of challenging network conditions and strict service requirements urgently presses to deploy the 5th generation of mobile communication systems (5G).

Nowadays, network operators, automotive industry, and service providers work closely together in order to provide a fruitful variety of vehicular services to their users, promising high levels of QoS, which reflects on the QoE. In particular, some of these services are safety-related (e.g., emergency electronic brake warning, lane change warning, and forward collision warning), non-safety (e.g., traffic information systems), and infotainment (e.g., peer-to-peer gaming, Internet Protocol Television (IPTV), Internet content sharing, and video streaming) [24, 28, 100, 101, 102]. Since the expectations towards vehicular communications are increasing, and ultra-low latency is a primary and critical concern for autonomous driving as an ultimate goal, there is an urgent need to leverage on emerging technologies such as 5G and MEC to facilitate the performance of various vehicular applications (Fig. 3.1). These applications, as any other 5G MEC applications, can be presented as a Service Function Chain (SFC) consisted of the VNFs that are loosely-coupled via interconnecting virtual links. Given that such vehicular applications are often resource-hungry (collecting and processing a huge amount of data from distributed sources), the SFC needs to be configurable upon changes in traffic and KPIs (e.g. latency, bandwidth, etc.), making it suitable for achieving low latency and high resource utilization. Thus, paving the way toward programmable and virtualized future communication networks, the incorporation of MEC platform into the SDN/NFV-based 5G networks brings flexible support to applications with diverse and stringent requirements, in terms of extremely low latency, high data rate, and high reliability [103, 104, 33]. Furthermore, with the ubiquitous support of SDN and NFV, MEC converges communication networks toward providing cloud-computing and diverse resources closer to the end-users, i.e. at the edge of RAN [33].

By jointly considering strict requirements for 5G networks, such as greater coverage, end-to-end latency less than 1ms, massive connectivity, and massive capacity, accompanied by enormous heterogeneity in network resources, technologies, vendors, operators, vehicles, etc., traditional manual network management becomes impossible to scale and to maintain. It is an utmost challenge to achieve high QoS and QoE without sophisticated management and orchestration [34], which are, at the same time, compliant to the standards. Hence, such heterogeneity presses an urgent need for transition from a poorly scalable network management to its automation. In short, bringing 5G and MEC to vehicular networks reduces data transmission time [37] for latency-sensitive use cases such as autonomous driving, but requires automated network management in order to cope with aforementioned heterogeneity.

Therefore, in this Chapter, we study the automated closed-loop life-cycle management and orchestration of network services and resources within MEC, based on the three inter-coupled activities, i.e., orchestration, control, and monitoring (as shown in Fig. 3.2). The recent advances in SDN and NFV aim to facilitate network management automation to a great extent when incorporated in MEC architectures. In particular, SDN and NFV bring more flexibility, and programmability to wired and wireless communication networks, while enabling higher resource utilization, and lower costs [36]. Yet, the full potential of such synergy is



(a) A high-level architecture of MEC-enhanced vehicular networks (single domain).

(b) Multi-domain vehicular networks.

Figure 3.1: Management and orchestration in MEC-enhanced vehicular networks.

still to be discovered [34].

The Fig. 3.1a illustrates a high-level architecture of MEC-enabled vehicular networks from a single domain perspective, while in Fig. 3.1b we can see how the same setup looks like if vehicular networks span multiple edge domains. In particular, a single domain spans the vehicles themselves (having On Board Units (OBUs) equipped with sensors), RAN, edge, and the NFV MANO entity. Providing storage and computational resources at the edge, MEC is intended to reduce latency for mobile users (i.e., vehicles) by utilizing more efficiently the mobile backhaul as well as the core networks [33]. The MANO manages and orchestrates MEC servers and the services deployed and running on top of these servers, and finally, the core network. The position of MEC platforms enables using resources exposed at the network edge to host, and to deploy numerous vehicular applications that can be easily instantiated, and terminated in a dynamic way. Furthermore, due to the opportunities to deploy vehicular services in a lightweight manner, MEC can also enable a migration of services from one machine that hosts the service to another if it is efficiently managed and orchestrated by a suitable MANO platform. In this way, a service migration tackles the low-latency requirements, since a new placement of services will ensure that low-latency can be achieved and maintained by responding to the vehicle movements in a proactive way. The bottom level of the overall network snapshot depicted in Fig. 3.1a presents an in-vehicle infotainment use case [100, 101, 102] in which vehicles exploit Content Delivery Network (CDN) as a service, with cache CDN servers placed within MEC in order to decrease the overall latency in accessing popular websites (e.g., Google maps). Such decentralized cloud architecture is going to be a cornerstone for vehicular communications, providing low latency services tailored to various 5G automotive use cases (e.g., active driving safety assistance, road traffic monitoring, cooperative maneuvering, in-vehicle infotainment, emergency situations, etc.). Simultaneously, this cloud architecture also assists in the offload of heavy computational tasks from autonomous vehicles to the edge.

In Fig. 3.2, we illustrate *orchestration*, *control*, and *monitoring*, as three simultaneous and

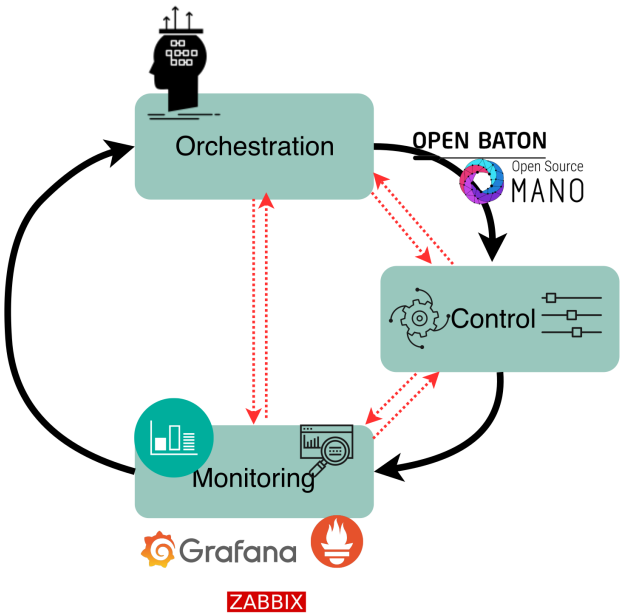


Figure 3.2: The closed-loop life-cycle management of network services.

intertwined processes. In particular, *orchestration* is backed by SDN and NFV in providing: i) automation, ii) coordination, and iii) managing, of deployments and operations of network services. The role of SDN is to provide connectivity, and to keep a centralized abstract view of the network topology [36, 105]. On the other hand, NFV is in charge of managing the network functions. With both support of SDN, and NFV, orchestration enables network services to be automatically deployed and managed [39]. In terms of *control*, we introduce existing MANO tools and exploit their control features. Finally, *monitoring* provides valuable input about available resources and network status to the orchestration entity, which can make decisions upon network services in a proactive and timely manner. Based on the decision made by an orchestration entity, the control is in charge of tweaking the network service configuration, and performing resource re-allocation.

The synchronization between these three interconnected processes (i.e., orchestration, control, and monitoring) is essential to enable automation for the resource and service management in strongly heterogeneous environments. Therefore, in this Chapter we also map such closed-loop life-cycle automation onto the existing NFV MANO systems, and present perspectives on their incorporation within MEC-based vehicular networks.

As a result of studying the most important KPIs for automated management and orchestration, there are two major groups, i.e., **feature-based** and **operational KPIs**, which can be used to benchmark existing MANO solutions for supporting delay-sensitive vehicular applications. Accordingly, we summarize the contributions of this Chapter as follows:

1. We map the architecture of MANO solutions to the *closed-loop life-cycle management and orchestration*, pointing at its clear articulation in the research and industry fields.

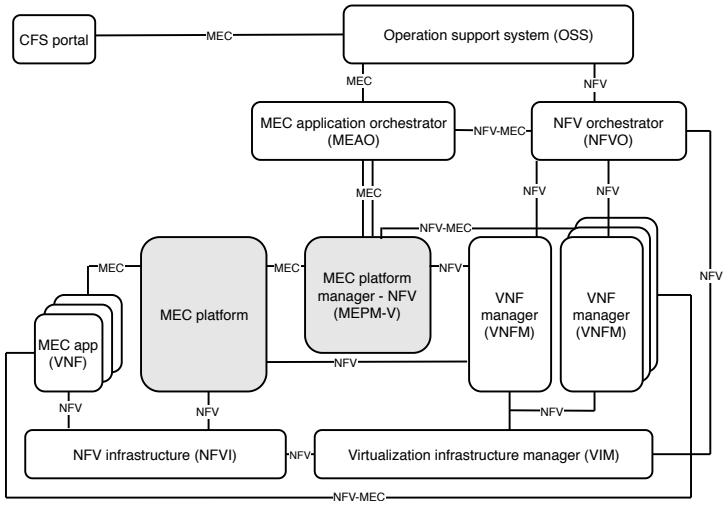


Figure 3.3: ETSI NFV MEC architectural framework.

2. *A feature-based analysis*: Taking into account the feature-based KPIs, i.e., key features of MANO tools (e.g., required resources needed to run a particular MANO, number of life-cycle management operations that MANO can effectively perform, etc.), we present a comprehensive overview of several MANO tools, developed within different research projects that recently increased the interest in network service orchestration. Due to their compliance to ETSI standards, and lightweight deployment opportunities, we see OSM and Open Baton as suitable solutions for MANO operations within the resource-constrained network edge. Therefore, we bring their extensive performance analysis conducted in a real testbed setup as our next contribution.

3. *A performance analysis*: One of the common operational KPIs that is used to measure performance of MANO solutions is an overall instantiation delay. In particular, it refers to the time needed for a MANO solution to successfully instantiate a fully operational network service. Therefore, based on this KPI we evaluate the performance of Open Baton and OSM in a testbed environment, and discuss their comparison while providing instructions on their deployment in vehicular networks based on 5G and MEC. Besides this comparison, we also showcase how different VIM systems affect the performance of orchestrators. This extensive performance analysis is obtained in the high-performance testbed, mimicking the realistic features of edge computing in vehicular networks.

This Chapter is organized as follows. A thorough description of the closed-loop life-cycle management of network services in MEC-based vehicular networks is presented in Section 3.1. This is followed by an in-depth analysis of the features of existing MANO tools in Section 3.2. In Section 3.3, we first present the performance analysis of Open Baton and OSM, and then analyze the impact of different VIM systems on both Open Baton and OSM.

3.1 The closed-loop life-cycle management of network services in MEC

Within the confines of this section, we present our first contribution, i.e., we present our vision of automated closed-loop life-cycle management of network services and resources, and map ETSI NFV MEC framework [106] to this closed-loop. The demand for transition, from a traditional manual network service management toward automation, is described in terms of: i) need to cope with strong heterogeneity in network resources, technologies, vendors, and operators, ii) achieving ultra-low latency to fulfill strict requirements for various automotive use cases (such as active driving safety assistance, road traffic monitoring, cooperative maneuvering, in-vehicle infotainment, emergency situations, among others), as high levels of other QoS and QoE parameters for vehicular applications, and iii) being less prone to dynamic changes in mobile data traffic and radio conditions caused by high speed mobile users (i.e., vehicles). The consolidation of all the aforementioned strict requirements is a challenging task, pressing an urgent need for automation of network service and resource management and orchestration. As illustrated in Fig. 3.2, we present this phenomena in the form of closed-loop life-cycle management of network services as an essential synergy between: i) *orchestration*, ii) *control*, and iii) *monitoring*.

Although separated, these three branches are exceedingly dependent on each other, with the ultimate goal to facilitate the whole network service management process.

In particular, to make reliable decisions upon vehicular and network services, it is inevitable for the orchestration entity to receive a real-time monitoring report from the monitoring entity. Furthermore, control entities that implement orchestration decisions have to consider monitoring input in order to track the changes, and to tweak the configuration of network services based on these changes. Thus, to be able to extract the potential of each process, and to enable automation of MANO operations in MEC and 5G, the synchronization of these parallel, but interconnected, processes is inevitable. In this section we present each of these processes forming the closed-loop, and discuss the importance of their particular share in the automation of network service life-cycle management.

3.1.1 Orchestration and Control

In order to facilitate MEC's incorporation into upcoming new generations of communication networks, and to enable better understanding of service and resource orchestration, ETSI formed an Industry Specification Group (ISG) to create a standardized and open environment, which enables the efficient and seamless integration of diverse applications from different vendors, service providers, and third parties [10]. ETSI NFV ISG defines NFV architectural framework, which is presented in Fig. 3.3, altogether with the constituent elements and reference points needed for hosting applications within MEC platform. From the automated closed-loop life-cycle management perspective, such architectural framework is depicted in Fig. 3.4, illustrating how we grouped the essential architectural elements into the orchestration, control, and monitoring categories. Therefore, due to their coexistence within ETSI framework, as well as their strong interdependence, the impact of orchestration and control is jointly discussed in this section.

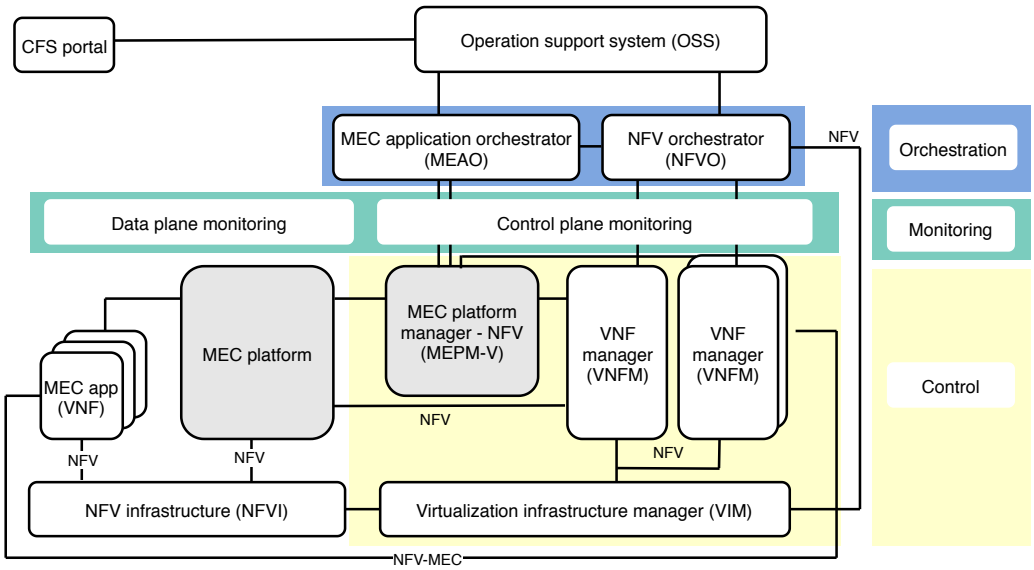


Figure 3.4: The closed-loop life-cycle managment of network services mapped to ETSI NFV MEC architectural framework.

The two main components of ETSI NFV MEC architecture are NFV Orchestrator (NFVO) and VNF Manager (VNFM), mutually assembling a so-called ETSI NFV MANO [107]. The NFVO, therefore, entails the orchestration functions, while VNFM stands for the control entity in charge of the life-cycle management of VNFs (i.e., VNF instantiation, scaling, terminating, etc.), as building blocks of the network services. Following the orchestration decisions and instructions provided by NFVO, VNFM manages all network service instances (i.e., VNFs) running in MEC, while VIM represents the management system for NFVI that is used for instantiation and operation of network service. To be more specific, the roles of VIM are: a) performing allocation, management, and releasing of virtualized resources, b) preparing the underlying NFVI to run software images as a base for the required VNFs, and c) collecting fault reports and performance measurements about virtualized resources.

The advantage of this open source architecture lays in facilitated implementation of an NFV architecture, increasing the likelihood of interoperability among diverse NFV implementations. The last is particularly important to emphasize, since different MEC platforms comprise several virtualized and physical resources, diverse services, and applications of various stakeholders. In such strongly heterogeneous environment, interoperability plays a crucial role, which can be assured only by following the standardization guidelines and recommendations.

3.1.1.1 Orchestration

The orchestration comprises processes of automation, coordination, and management of deployment and operation of network services [39]. In particular, the NFV architecture and orchestration framework proposed by ETSI establishes the following three domains: i) VNFs, as software defined network functions, ii) NFVI, consisted of hardware and software

components for deploying VNFs, and iii) NFV MANO providing organization and management of NFVI, which is responsible for the life-cycle management of VNFs, i.e., network services [33, 104]. As Fig. 3.4 clearly depicts, the orchestration in such architectural framework spans two different blocks, i.e., MEC Application Orchestrator (MEAO) and NFVO, which perform life-cycle management operations of MEC applications and network services, respectively.

In their survey on network service orchestration, de Sousa et al. [39] claim that the foundations of orchestration are routed back to the three enabling technologies, i.e., SDN, NFV, and cloud computing, where the SDN is in charge of enabling connectivity, NFV of managing the network functions, and network service orchestration governs all the deployment processes of the end-to-end network service. According to the study presented in their survey [39], de Sousa et al. classify orchestrators based on their functional scope, as follows: i) *service orchestrator* – carries out service composition/decomposition, ii) *life-cycle orchestrator* - manages the workflows, processes, and dependencies across service components, and iii) *resource orchestrator* - maps service requests to resources, either virtual or physical. Another classification is provided based on the operational scope of the orchestrator, where the *domain orchestrators* have an absolute control over all resources that belong to their unique domains, but being limited to the administrative boundaries. On the other hand, *multi-domain orchestrators* have a broader scope but are therefore more complex, enabling end-to-end service orchestration while spanning different administrative domains [39].

According to Taleb et al. [33], the true impact of MEC paradigm relies on the service orchestration capabilities as well as on the interaction with network architecture. Being aligned with ETSI NFV framework, MEC framework (Fig. 3.3) includes virtualized infrastructure, as well as applications, and VNFs deployed on top of it. The service-related attributes such as resource allocation, service placement, edge selection, and reliability, are of particular relevance for the efficient orchestration [33]. In the context of resource allocation, Taleb et al. [33] provide an overview of research efforts to study how the efficient resource allocation strategies impact the overall process of orchestration. A strongly heterogeneous pool of resources (virtualized and physical) is present within MEC platforms, being allocated to serve various services and applications installed on top of the platforms. Hence, it is expected that the brain of the orchestration process – i.e., orchestrator, takes care of efficient resource utilization in order to meet stringent service requirements on .g., latency, service reliability, and throughput, such as those in vehicular networks. In the context of vehicular applications, MEC service/VNF placement, including the MEC server selection over different platforms, is an utmost challenging task due to high speed mobility and use-case-dependent service deployments. It means that different stakeholders might be included in the service design and deployment, which depend on the specifications required by different use-cases. For instance, in cooperative maneuvering or mission-critical use-cases, multiple vehicles might be served by one or multiple MEC servers. The latter requires multiple instances of service being instantiated on each edge server, which is suitable for hosting application. Therefore, the orchestration is in charge of managing all service instances among different MEC platforms, in a manner which enables achieving corresponding QoS, QoE, and resource utilization.

Looking at the orchestration from a broader perspective, a baseline group of management and orchestration operations comprises service instantiation/placement, scaling, migration/relocation, and termination, and we provide more information about them as follows.

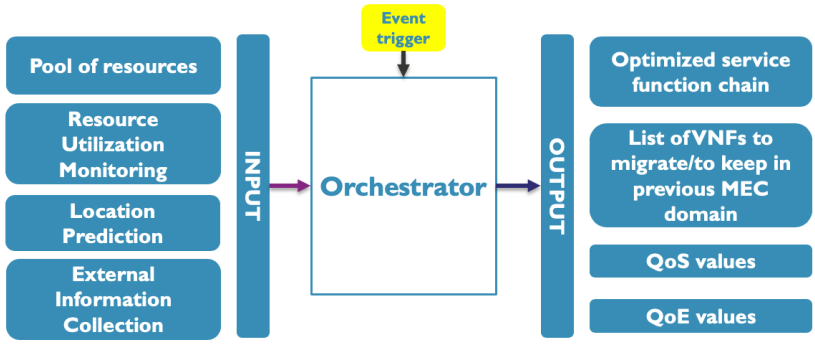


Figure 3.5: A high-level overview of an automated on-the-fly VNF placement and migration.

VNF instantiation/placement The VNF placement process consists of the two following phases: i) the composition of SFC, and ii) the SFC embedding to the substrate network which consists of the physical hosts. Although the phases are executed in a sequential manner, they cannot be detached and the overall process of VNF placement has to be coordinated. The process of SFC embedding is performed in the following order: i) determining the traffic paths, ii) reserving bandwidth on the links which constitute the determined traffic paths, iii) instantiating VMs or containers at different nodes on the determined paths, and iv) installing VNFs on the instantiated VMs or containers. Regarding more efficient resource utilization, the VNFs can be shared among different SFCs, depending on the VNF functionalities and specific limitations which are mostly defined for the security reasons. The resource management task is to determine the amount of resources which will be enough to obtain satisfactory level of resource utilization, and in order to maximize the resource utilization efficiency, VNF placement management should exploit the sharing potential. Although VNF placement has already increased the awareness about importance of resource utilization reduction, now with the edge computing it is even more interesting to inspect the impact of network conditions on the placement.

VNF migration, scaling, and termination These three operations are considered as runtime operations, as they are performed by orchestrators during the VNF/service runtime. The scaling procedure is in charge of assigning more or fewer resources to the running VNF, in case of scaling up/out and scaling down/in, respectively. The orchestrators usually make decisions when and how to perform scaling based on the monitored performance, i.e., taking into account a pool of available resources, resource utilization, and vehicle location, as illustrated in Fig. 3.5. The same applies to the VNF/service termination, which performs deletion of the VNF/service, thereby releasing the reserved computing and network resources. Furthermore, the VNF/service migration/relocation could be also considered as scale out operation that stretches multiple MEC hosts, and as such, it requires more discussion. In the context of multiple edge domains, VNF migration/relocation implies relocation of network service, i.e., VNF chains, from MEC servers in one domain to corresponding MEC servers in another. This type of migration has to be real-time in order to avoid potential disruptions in service continuity, which will result in undesirable effects (i.e., degradation of the QoS and QoE). Therefore, the migration of the VNF chain is necessary when resource requirements of VNF exceed the threshold of either the physical node or the physical

link where VNF is deployed to [108]. The service migration is an essential process within managing network services, which are consumed by users that move through the adjacent geographical areas. In general, service migration means migrating an ongoing service or application from one edge/cloud host to another, while state exchange refers to copying the state of the service/application from the source host to the target host. As stated by Wang et al. [109], the service migration should deal with the decision on whether the particular service should be migrated or not, and if yes, then with the decision on which host this service should be migrated to, thereby taking into account the overhead that this migration brings, as well as the QoS requirements. Since the synergy of 5G and MEC promises the ultra-low latency (i.e., 1ms-10ms) and high capacity (i.e., above 100 Mbps), these stringent QoS requirements press an urgent need for network service management systems to follow the user mobility, and to place network services always at the most suitable MEC platforms [110, 111]. Although both processes are initiated by user's movement from one area to another, the handover and service migration should be differentiated, and treated differently. In their latest survey, Wang et al. [109] also highlight the differences between these two processes, which are summarized as follows:

- *Amount of data to be transferred*: the data to be exchanged between source and target hosts during handover usually contains only the signal messages between UE and gNB, or two gNBs that handle the handover process, while in the case of service or state migration the memory data and/or application data image messages should be transferred, burdening the system with traffic that is usually a way larger than signaling,
- *Technology diversity*: in cellular networks, the handover is always performed between two neighboring cells with the same technology, while service migration is independent of the technology, and usually occurs in heterogeneous environment with different network topologies and technologies in edge domains,
- *Triggering the process*: while the handover is required anytime UE exits the coverage area of a particular gNB, in case of service migration, UEs can still exchange data with remote edge server, thereby bringing additional complexity in the whole system.

Under the umbrella of service migration, there are the following practical concepts that are widely studied and adopted in industry and research: i) VM migration, ii) container migration, and iii) stateful process migration [112]. As an application is usually realized in the form of a set of execution environment (e.g., operating system) and services that are required for an application to run, the aforementioned concepts differ in the components of the overall application that are migrated [112]. Hence, in case of VM migration, all application components need to be migrated from source to the target host, which due to the amount of data takes more time. In addition, there are different types of VM migration, such as cold migration, pre-copy live migration, and post-copy live migration, which are elaborated by Abdah et al. in [111]. On the other side, in case of container migration, the execution environment is not migrated but only the service (i.e., stateless and stateful). Finally, in the case of stateful process migration, only the stateful processes in the application are migrated from one host to another. As network edge is usually characterized by constraints in both network and computing resources, service migration also needs to be network and resource-aware. Therefore, Horri et al. [112] studied the concept of separating stateless

and stateful processes inside an application, allowing them to talk to each other via inter-process communication channels that, once stateful processes are migrated, need to be re-established on the destination server. Furthermore, according to Addad et al. [110], this migration of stateful processes can be obtained in two manners: i) stateful service migration with predefined path, in case the system can anticipate the source and the target MEC nodes for any migration along the way of the user, and ii) stateful service migration based on undefined path, which is a more generic approach since service providers usually do not know the movement patterns of their users [110]. While in the first case, the need for service migration can be anticipated and thus preemptively triggered and performed, in the second case it becomes impossible and Addad et al. [110] study and present the fast and efficient migration process with a shared file system/pool that stores the container's file system. Finally, concerning the migration costs, Strunk provides an overview of all contributing factors to the overall service migration cost in [113]. The costs that vastly influence the service performance are the total migration time, the downtime, the energy overhead, but also the impact on the performance of VMs after migration, such as execution time and throughput of processes running inside a VM during migration [113]. The total migration time is studied and evaluated in different migration approaches [113, 114, 110, 112, 111] and it highly depends on the total amount of memory that has to be transmitted from source to target hypervisor/host and average link speed between these hosts [114], but also on the CPU resources of the source host due to the increase of processing cycles caused by migration.

3.1.1.2 Control

The essential control blocks included in ETSI NFV architectural MEC framework are illustrated and emphasized in Fig. 3.4. As already stated in previous section, VNFMs are responsible for the VNF life-cycle management tasks including, for instance, its instantiation, scaling, pausing, restarting, and termination. However, VNFM is also in charge of reporting the VNF states to NFVO, so it can promptly react to changes, and make decisions on VNF placement and relocation. More so than ever, the dynamic changes in network traffic and service request patterns require continuous management of services, in terms of allocating more resources, VNF scaling up or down, releasing unnecessary resources, and terminating, with an ultimate goal to achieve or maintain satisfactory level of QoS, QoE, and resource utilization.

Besides VNFMs in the control entity shown in Fig. 3.4, there is a MEC platform manager which: i) manages installed MEC applications (e.g., vehicular applications), including informing orchestrator of relevant events from applications, ii) provides element management functions to the MEC platform, and iii) manages application rules and requirements (such as service authorization, traffic rules, etc.) [115]. Another role assigned to the platform manager is to control fault reports and performance measurements about virtualized resources, which are all collected by VIM and forwarded to the platform manager for further processing [115].

To enable development and deployment of VNFs, and MEC applications, controlled by VNFM and MEC platform manager, virtualized infrastructure consisted of computing, storage, and networking resources requires proper control as well. Therefore, VIM performs

the allocation, management, and releasing of these resources, and prepares the underlying NFVI to run software images as basis for the required VNFs. As already mentioned, VIM also collects and reports performance and fault information about resources, delegating the reports to VNFMs. Importantly, once when it is supported by MANO systems, service relocation/migration will be performed by VIM [115].

3.1.2 Monitoring

As we identified monitoring as one of the three crucial segments of closed-loop life-cycle management of vehicular services in 5G networks enhanced by MEC, this section summarizes the main research efforts towards monitoring network services to improve management and orchestration efficiency. In general, the overall monitoring process has to ensure that each network service is running properly, by extracting the critical information from the physical or virtual nodes (e.g., network functions, links, and user equipment), and sending important notifications to the orchestration and control entities. It comprises data collection and information extraction, which are directly performed by monitoring entity shown in Fig. 3.4. The extracted information is further leveraged by orchestration plane which makes corrective decisions. Afterwards, the control entity performs the actions implied from the orchestration decisions, which might include resource re-arrangement, VNF/service migration, scaling, and terminating.

The project 5GTango [116] has recognized the importance of having an adequate monitoring tool to be embedded into automated management system in 5G networks. Therefore, the project consortium [117] has identified several constraints of currently available monitoring tools, which limit their usage in 5G networks, as follows:

- intrusiveness for short-lived network function instances
- not being able to follow the pace of dynamic management
- not covering the requirements for both container-based and hypervisor/VM-based network function deployments
- not being suitable for collecting data from different cloud environments.

Taking into account aforementioned characteristics that constrain monitoring of network services, incorporation of a monitoring tool with general purposes into the closed-loop life-cycle management of MEC-based vehicular services is not a straightforward task. Although theoretical, an effort to approach this problem is presented by Celdran et al. [40]. In their study of automatic monitoring for 5G networks, Celdran et al. [40] note that monitoring has to be included within automated management of 5G services, since otherwise managing monitoring of network services would be impossible to perform due to the enormous number of connected devices and their high mobility. The authors provide an important aspect for isolating the information which needs to be monitored, in order to provide necessary input for network service life-cycle management and orchestration with an ultimate goal to improve QoS and resource utilization.

There are two distinct types of information to be monitored: i) Data-related information (DRI) such as information contained in network flows, and ii) Control-related Information (CRI) – such as users' mobility, network infrastructure location, number of active users, and percentage of CPU and storage consumed by the network service. It is particularly important to monitor the CRI type of data, as it provides valuable input about the users who are consuming the service, and thus, the orchestrators can tailor their decisions and operations to improve service quality at users' side. For instance, if orchestrators take into account the current locations of vehicles and/or average CPU load on the distributed edge domains, they can make proactive decisions that will re-configure the service and improve its performance for the users. In addition, there are various mechanisms to measure this type of data, such as leveraging on the interfaces towards 5G Core functions, or Kubernetes mechanisms for monitoring computational load on the allocated NFV infrastructure. Therefore, Celdran et al. [40] propose a solution which incorporates monitoring into the architecture oriented toward 5G networks, which integrates SDN and ETSI NFV architectural proposal. In order to adequately manage the monitoring process, they propose to monitor control and data plane separately (Fig. 3.4). With a specific focus on the control plane, i.e., gathering CRI, the architectural components (e.g., VNFM, VIM, SDN applications, etc.) expose the information to CRI monitoring component, which therefore aggregates all the upcoming information and forwards it to the decision making entities. In such asset, the monitoring on the VNF level can be performed, tweaking resources allocated to each VNF based on the decisions made in orchestrator.

There are various monitoring tools available for different purposes, and for instance, cloud monitoring has a resourceful research background. However, all of these tools are customized to the specific types of VIM (e.g., OpenStack [118], Amazon Web Services (AWS) [119], VMWare [120], and OpenVIM [121].), making them dependent on the specific virtualized infrastructure, which is difficult to scale especially in such heterogeneous environment as MEC in 5G. For instance, the most popular monitoring solutions for OpenStack are Ceilometer and Nagios, which meter the data related to OpenStack resources such as compute, networking, and storage. In case of AWS, there is CloudWatch that monitors Amazon EC2 instances, Amazon RDS databases, and Amazon DynamoDB, and sets alarms with specific priorities based on the severity and importance of the information that is being monitored. Hamid and Shah [122] assess the performance analysis of the aforementioned types of monitoring tools, including vROPS that is used for monitoring VMWare resources. Their effort to integrate AWS monitoring support into the OSM orchestration tool is presented in [122], in which they elaborate on the idea to create an integral monitoring component which will consist of various plugins customized to different VIMs. In particular, they detail on how to create plugin for monitoring AWS resources, aiming to automatize the overall monitoring process by excluding the need for manual configurations. Such active monitoring of individual resources that belong to AWS cloud enables proactive and automated troubleshooting and self-healing of resources [122]. However, due to their strong dependence on the specific VIM types, the capabilities of available monitoring tools are limited, and therefore research in this field should be further intensified.

3.2 A feature based analysis of existing MANO tools

As our second contribution in the Chapter, this section presents an extensive feature-based analysis of existing open source MANO tools, which are widely recognized in both academia and industry circles. Through a thorough examination and study of the available documentation and research papers that tackle a particular MANO tool, we isolated key features that need to be taken into account when studying these tools. We find such analysis as notably important for the future research in the field of resource and service orchestration, because it provides a summarized information on the tools which are likely to be used in the real deployment, and can be used as guidelines for future extensions of existing orchestrators. Each particular feature is essential to consider, as it highly affects the performance of the tools and their ability to get customized to different experimental environments.

Based on the work provided by Taleb et al. and de Sousa et al. [33, 39], the open source tools that attracted significant attention in past few years are ONAP, Open Baton, Sonata (5GTango), OSM, Tacker, Cloudify, X-MANO, TeNoR, and Escape. Since the background information for each of these tools, such as the research projects in whose scope the tool was developed, is already presented in aforementioned work, here we do not present the specific project and tool details. Therefore, in Tables 3.3, 3.4, and 3.5, we map the feature types to their corresponding metrics for each MANO tool that we took into consideration, and the brief discussion based on each feature is presented as follows.

Resource footprint It embodies one of the fundamental requirements prior to experimenting with a MANO tool, because it presents the amount of resources (such as number of virtual or physical machines, RAM, number of vCPUs, storage, etc.) needed for the installation and proper work. To make the result comprehensible, we present three categories, i.e., *light*, *medium*, and *heavy*, and map the required resources to them as presented in Table 3.1. Concerning the resource footprint, the three categories presented within Table 3.1 can help readers to resolve where is a certain MANO solution positioned on the scale from being lightweight to resource-hungry. The categories are based on the number of virtual CPUs that each MANO solution requires for its proper work, as well as the optimal values of RAM and storage. For example, the *light* MANO solutions can be successfully deployed inside a VM on the host, while *medium*, and especially *heavy* solutions, in most of the cases require dedicated resourceful bare-metal servers to efficiently perform their tasks. In Table 3.3 and its extension (Table 3.4), the resource footprint is shown for each tool. It can be seen that ONAP is the heaviest in terms of all three resource components, which is expected due to its extensiveness, strong credibility, and relevance for the industry as well. On the other hand, Open Baton and OSM offer two installation possibilities, i.e., minimal and full, which differ in number of supported components (e.g., NFVO, VNFM, drivers for monitoring plugins, drivers for different VIMs, etc.). However, it should be noted that MANO tools that connect to VIMs such as OpenStack, require additional machines to install VIM, which in particular needs 4 vCPUs, 8 GB RAM, and more than 80 GB of disk space per se.

Messaging bus This specific component is essential for enabling either synchronous or asynchronous communication between different MANO components, offering message ex-

Table 3.1: Resource footprint categories for MANO tools.

	light	**medium**	**heavy**
number of vCPUs (N)	$2 \leq N \leq 4$	$4 < N \leq 8$	$N > 8$
RAM (R)	$R \leq 4$ GB	4 GB $< R \leq 8$ GB	$R > 8$ GB
storage (S)	$S \leq 20$ GB	20 GB $< S \leq 40$ GB	$S > 40$ GB

Table 3.2: Overview of messaging buses.

Messaging bus	**Message exchange**	**Message protocol**	**Queueing**	**Complexity**
RabbitMQ	synchronous/asynchronous	Advanced Message Queueing Protocol (AMQP)	via centralized node	low
ZeroMQ	asynchronous	ZeroMQ Message Transport Protocol (ZMTP)	decentralized	high

change in a reliable way. The overview of two widely used messaging buses that are also utilized within MANO solutions, i.e., RabbitMQ and ZeroMQ, is presented in Table 3.2. The Table 3.2 depicts the main differences between these two messaging buses in terms of the message exchange mode, message protocol, the mode of queueing, and their complexity. In particular, a complexity refers to the source code of the messaging bus, i.e., the number of lines of code needed to realize routing, load balancing, and persistent message queueing. As it can be seen from the Tables 3.3 and 3.4, the great majority of tools use RabbitMQ messaging bus, due to its powerful and flexible operation. RabbitMQ is an open-source general purpose message broker that implements a variety of messaging protocols, with Advanced Message Queueing Protocol (AMQP) among them. In MEC-based MANO case, RabbitMQ provides MEC applications with a platform to send and receive messages, connect to each other, and scale. It is performed through different versions of point to point, request/reply, and pub-sub communication style patterns, which enable publishers to send messages to exchanges (central nodes), and consumers to retrieve messages from queues [123]. Due to this simplistic operation mode which enables routing, load balancing, and persistent message queuing in terms of several lines of code, RabbitMQ is easy to use and deploy, and therefore, it is reasonable that most of the MANO solution developers opt for this messaging broker. However, it inevitably generates additional latency because of message queuing on a central node. In regards to that, ZeroMQ [124], engaged by Escape, is a lightweight substitute for RabbitMQ, as it specially addresses latency constrained networking scenarios such as autonomous driving. Due to the fact that ZeroMQ deploys messaging in a purely distributed manner, its design complexity is larger than in case of RabbitMQ, which does not necessarily mean that it makes it more complex to use. Thus taking into account the importance of low-latency for vehicular applications, decision upon messaging system should be taken with a prominent attention, studying and benchmarking both RabbitMQ and ZeroMQ to find a trade-off.

Infrastructure adaptation Under the term of infrastructure adaptation, we consider the capability of the MANO tool to adapt to different types of VIM. The more VIM drivers supported by tool, the more flexibility in experimentation and deployment is provided. This is significantly important since different VIM types are more or less complex than the other, and

Table 3.3: A feature based analysis of existing ETSI NFV MANO systems - part 1.

Feature type		ONAP	Open Baton	Sonata (5G Tango)	OSM
Resource footprint	**number of vCPUs**	heavy	minimal version: light full version: heavy	light	minimal version: light full version: light
	RAM	heavy	minimal version: light full version: heavy	medium	minimal version: light full version: medium
	storage	heavy	minimal version: light full version: light	heavy	minimal version: light full version: medium
Messaging bus		Microservice Bus/ Message & Data Routers	RabbitMQ	RabbitMQ	RabbitMQ
Infrastructure adaptation		VIM: OpenStack, Azzure, VMWare, and Wind River	VIM: OpenStack, Amazon, Docker, Test	VIM: OpenStack, Kubernetes, Sonata Emulator, WIM: Virtual Tenant Network (VTN) and Transport API (T-API)	VIM: OpenStack-based, VMWare, AWS, OpenVIM
Virtualization environment		VMs (currently), but VNFs hypervisor agnostic	VMs and containers	containers	VMs
VNF life-cycle operations		1. instantiation	1. instantiation	1. placement	1. modelling
		2. configuration	2. configuration	2. on-boarding	2. on-boarding
		3. elastic scaling (only horizontal - scale in/out)	3. starting	3. instantiation	3. NS creation
			4. stopping	4. scaling in/out	4. NS operation
		4. automatic recovery from resource failure	5. terminating	5. termination	5. NS finalization
			6. scaling-in		
VNF package	**VNF descriptor**	TOSCA, YANG	TAR, CSAR (TOSCA)	domain specific language similar to TOSCA and HOT	YAML-based documents
	VNF image	N/A	QCOW work in progress	N/A	QCOW
VNF healthy environment support		various packaging and validation tools available and integrated	no	yes	no
Integrated monitoring system		yes	no, connecting to various systems via plugin mechanism (Zabbix plugin)	yes (advanced real-time monitoring system)	no, plugins for different VIMs available (1. AODH/Gnocchi for OpenStack, 2. VMware vRealise Ops Update, 3. AWS CloudWatch), VNF monitoring-Grafana
Feature palette		1. deployment	1. deployment	1. life-cycle management of NSs, slices, VNFs	1. NS/VNF on-boarding
		2. configuration,	2 managing PoPs	2. management of SLA	
		3. monitoring	3. catalogue	3. performing VIMs, WIMs, end Endpoints	
		4. restart	4. marketplace	4. monitoring KPIs	2. lifecycle
		5. clustering and scaling	5. launching NSD	5. catalogue	
		6. upgrade	6. on-boarding NSD	6. specifying QoS requirements links	3. fault and performance management
		7. deletion			
Interfaces		Portal, Dashboard, Use case UI, External APIs, CLI	Dashboard (GUI), CLI	Portal (GUI), WEB interface, CLI	Dashboard (GUI), WEB interface, CLI
Operating system		Ubuntu	Ubuntu 14.04, Ubuntu 16.04	Ubuntu	Ubuntu 16.04

if diverse set of VIM drivers can be easily installed within MANO, it expands the possibilities to combine resources from different virtualized infrastructures. All studied tools support OpenStack, as a widely used software platform which offers a plethora of virtualized servers and other resources to customers. However, due to the increased complexity in configuring OpenStack to work with a particular MANO tool, the support for additional VIM drivers that can be easily configured (e.g., AWS) should be more accentuated and motivated.

Virtualization environment Despite the enormous popularity of Virtual Machines (VMs), the container-based virtualization is now gaining momentum, due to its capability to share the host kernel with user-space isolation. There is already a solid research conducted on

Table 3.4: A feature based analysis of existing ETSI NFV MANO systems - part 2.

Feature Type		Tacker	Cloudify	X-MANO	TeNoR	Escape
Resource footprint	**number of vCPUs**	medium	medium	N/A	N/A	N/A
	RAM	medium	medium			
	storage	heavy	light-heavy			
Messaging bus		RabbitMQ	RabbitMQ	RabbitMQ	RabbitMQ	ZeroMQ
Infrastructure adaptation		VIM: OpenStack and Kubernetes	VIM: AWS, Azure, OpenStack, Vsphere	N/A	VIM: OpenStack, Open Daylight	VIM: OpenStack
Virtualization environment		VMs and containers	VMs and containers	VMs	VMs	containers
VNF life-cycle actions		N/A	1. event-stream processing	1. creation	1. start	1. initiate/start/ stop NF
			2. metrics queueing	2. chaining	2. stop	2. connect/ disconnect
			3. aggregation		3. restart	3. NF to/from switch
			4. analysis, etc.	3. deletion	4. scale-in	
					5. scale-out	
VNF package	**VNF descriptor**	TOSCA	TOSCA	VNF manifest: JSON file, multi-domain NS descriptor: YAML	HOT	YANG
	VNF image	N/A	QCOW	N/A	N/A	N/A
VNF healthy environment support		no	no	no	no	no
Integrated monitoring system		no, drivers for Aodh, and Ceilometer	yes	no, plugin for Zabbix	no, plugin for VIM monitoring and Apache Cassandra	no
Feature palette		1. VNF Management: VNF Catalog and VNFM	1. uploading and deleting blueprints	1. VNF catalogues	1. NS/VNF Monitoring	1. SDN domain manager
			2. keep a directory of blueprints			2. Internal domain manager
			3. create multiple deployments for each blueprint,		2. NS/VNF provisioning	
		2. NFV Orchestration: VIM Management, VNFFG Catalogue, VNFFG Manager, NS Catalogue, NS Manager	4. execute workflows	2. NS management panel		3. Remote domain manager
			5. execute healing and scaling		3. Service Mapping	
			6. view application's topology			4. OpenStack domain manager
			7. retrieve events	3. statistics panel (visualize and export collected monitoring information)		
			8. utilize plugins			
			9. view metrics		4. SLA Enforcement	5. Universal Node Domain manager
			10. search logs			
Interfaces		Horizon and CLI	CLI, WEB UI	Customer portal (GUI)	N/A	REST-API, GUI
Operating system		1. CentOS, Redhat, Oraclelinux (source and binary images), 2. Debian and Ubuntu (only source images)	1. RHEL/CentOS 6.x 2. RHEL/CentOS 7.x 3. Ubuntu 14.x/16.x/18.x 4. Windows 2008 and later	Ubuntu 14.04 LTS, Windows 8.1 and Windows 10	Ubuntu 14.04	Ubuntu 16.04

Table 3.5: A feature based analysis of existing ETSI NFV MANO systems - part 3.

Feature type		ONAP	Open Baton	Sonata (5G Tango)	OSM	Tacker	Cloudify	X-MANO	TeNoR	Escape
ETSI NFV MANO Compliance	**NFVO**	yes	yes	yes	yes	yes	not fully	yes	yes	yes
	VNFM	yes	yes	yes	yes	yes	not fully	yes	yes	no
Multi-domain support		yes	no	no	no		no	yes	yes	yes
Multi-tenancy support (Network slicing)		yes	yes	yes	yes		no	N/A	N/A	N/A

capabilities of both VM and container-based virtualization, studying the benefits and limitations of both [39, 33]. Tackling the resource availability within the MEC platforms, which is limited comparing to the large and resourceful data-centers, the lightweight virtualization, and orchestration solutions for small-size programmable devices are required. Delivering a lightweight deployment of services and applications, containerization seems to be the best candidate for deployment of emerging 5G technologies such as NFV and MEC [104]. Therefore, the MANO tools with support for a container-based virtualization are considered as profoundly interesting for future MEC-oriented research. The aforementioned enables orchestration and management of the latency constrained applications, placed and deployed within the edge of the vehicular networks.

VNF life-cycle operations Depending on the type of the MANO tool, a certain number of life-cycle management operations is supported. Keeping in mind ONAP's superiority and extensiveness comparing to other tools, a support for a plentiful set of operations is expected. If we tend to approach the study of tools with lower complexity and *lighter* installation, most of the remaining tools provide support for number of operations of similar scale. Importantly, all of the tools enable three fundamental actions, i.e., instantiation, scaling, and termination. In particular, instantiation and on-boarding operations are usually tightly coupled. On-boarding means transferring appropriate image file altogether with VNFD and NSD, from NFVO to VIM via VNFM. In that phase, VIM allocates resources required for such VNF and network service, based on the specified flavor. On the other hand, instantiation is represented as a phase of booting-up a system based on the received image, and installing all dependencies stated in descriptors, which are needed for VNF or network service to run properly. In case of scaling, more resources are needed than it was initially allocated by VIM. Thus, based on the instruction from NFVO, VIM re-allocates resources, and in case of termination it releases the resources.

VNF package A VNF package includes a corresponding VNFD that will be used to describe a VNF, as a part of the service chain that orchestrator aims to launch on top of the virtualized infrastructure. Besides VNFD, which provides a broader communication compatibility among operators, there is an NSD as well, containing description of the whole network service. Depending on the tool, these descriptors are usually written following some of the well-known standards, such as: Topology and Orchestration Specification for Cloud Applications (TOSCA), Yet Another Next Generation (YANG), and Heat Orchestration Template (HOT). For instance, TOSCA is a standard used to specify services and their relations on a cloud computing view, while YANG represents a data modeling language for configuration and state data manipulated by the network configuration, designed by IETF. Similarly to TOSCA, HOT in particular, describes the resources and the relationship among them. However, being much more generic and able to automate any application production process, TOSCA is widely used for describing VNFs and network services. Nevertheless, given the broad support and availability of all three standards, we consider TOSCA, YANG, and HOT suitable for the orchestration solutions that we tackle in this paper. Finally, besides descriptors, a VNF package usually includes a VNF image which needs to be available on the corresponding VIM, so that Element Management System (EMS) entities are provided with an adequate image type for launching VNF-customized VMs.

VNF healthy environment support This feature is quite specific since it is only available in ONAP and Sonata, representing incorporation of VNF self-healing capabilities such as those provided by integrated validation tools. In case of large-scale usage of the tools in industry and production, such capability is essential.

Integrated monitoring system Recalling the closed-loop life-cycle management, which was presented within Section 3.1, and mapped to the ETSI NFV MEC architectural framework, there is a huge potential in integrating a monitoring system into the MANO solution. Such possibility decreases the delay in communication between monitoring and orchestration and control entities, therefore providing real-time information gathered from the measurements. Although some of the tools (e.g., ONAP, Sonata, and Cloudify) incorporate a tool-customized monitoring systems into their architectures, most of the studied MANO solutions still require installing plugins for external monitoring (such as Grafana, Zabbix, etc.).

Feature palette It comprises different capabilities that MANO tool can provide to the users once it is properly installed. The palette is usually reached through some tool-specific Graphical User Interface (GUI), and in most of the cases it shows the actions that can be taken during the VNF life-cycle management.

Interfaces Almost all of the encompassed MANO tools provide work on the resource and service orchestration, specific component configuration, actions from the life-cycle management set, and various activities from the feature palette, through both GUI - usually represented as a dashboard, and a Command Line Interface (CLI). Understanding of all the processes of VIM registration, creating VNFDs and NSDs, on-boarding VNFs, launching network services, etc., is facilitated by providing a corresponding GUI, as it is more representative than a usual CLI. Although the installation of each tool must be obtained through the CLI, representing the feature palette within a GUI-based dashboard is a plus.

Operating system This feature only reflects the requirements based on the fundamental operating system, required for installation and proper work of the MANO tool.

ETSI NFV MANO compliance In general, in order to expand the exploitability of any software tool, whether it is MANO or not, the standardization plays a key role as it assures that the tool meets certain requirements that guarantee the proper work in various conditions. Having ETSI NFV MANO framework (Fig. 3.3) as a reference, it is unlikely that a tool with no proper compliance will be considered as a candidate for the resource and service orchestration in MEC-enhanced vehicular networks, because ETSI has a leader role in standardizing NFV and MEC. The necessity for standardization in aforementioned context is reasonable, especially because of the heterogeneity in MEC platforms. Therefore, although developed and deployed by different vendors/operators/application designers and developers, various MEC platforms and applications can be consolidated and able to cooperate if the standardization requirements are met.

Table 3.6: Types of service function chains.

Network Service (Service Function Chain)	Number of VNFs in the chain	VNFs
SFC_1	1	VNF_1
SFC_2	2	VNF_1, VNF_2
SFC_3	3	VNF_1, VNF_2, VNF_3
SFC_4	7	VNF_1, VNF_2, VNF_3, VNF_4, VNF_5, VNF_6, VNF_7

Multi-domain support The multi-domain capabilities represent a strong contributing factor to filter the orchestration solutions, being characterized by capabilities to establish a connection with MEC from the other domain using technologies such as OpenVPN, and to enable communication among the resources in different administrative and technological domains.

Multi-tenancy support Due to the ubiquitous popularity of network slicing paradigm, it is important to be able to allocate different slices of network resources to different tenants with specific QoS and QoE requirements. In the context of our work, the tenants could be verticals that use edge services in their operation. Depending on the type of vertical, edge services and EdgeApps have different service requirements, and it is important that the orchestration systems are capable to recognize different requirements and to accordingly tailor service and EdgeApp deployments.

3.3 A performance analysis of existing MANO tools

Linked to the third contribution of the Chapter, this section shows a performance analysis of two open source MANO solutions, i.e., Open Baton and OSM, aiming at inspecting their suitability for orchestrating realistic latency-sensitive vehicular applications, and we also study their performance analysis with reference to the VIM systems they use. First, we outline the overall experimentation setup by presenting: i) the type of network service that we used for testing, ii) the metrics that we defined in order to evaluate the performance of MANO solutions, and iii) description of testbed that we used for assessing a performance evaluation. Second, we present the experiments that we designed to evaluate the performance of MANO systems, the results that were collected during the measurement of the KPIs presented in 3.3.2, and then discuss those results and point at the clear articulation of incorporating these MANO tools into the framework of automated closed-loop life-cycle management of vehicular services.

3.3.1 Network service

As a service that needs to be dynamically instantiated, for the experimentation we have chosen a CDN as a Service (CDNaaS), which could be mapped to the group of infotainment services within a vehicular context (e.g., loading Google maps with reduced latency), thereby investigating whether MANO tools are capable to enable dynamic service creation

Table 3.7: Overview of metrics.

Metric	Definition
Overall instantiation delay (OID)	The overall time needed for a network service to be on-boarded and instantiated on top of the NFV Infrastructure (NFVI).
Overall termination delay (OTD)	The overall time needed to release resources when terminating the service
CPU utilization	The average usage of CPU processing resources, i.e., the amount of work with which a MANO solution burdens the CPU of the underlying host.
RAM utilization	The average allocated memory needed for a MANO operation.

and management. As Taleb et al. presented in [125], CDNaaS represents a service instance of virtual CDN, with aim to strategically instantiate and place CDN VNF instances over the cloud/edge nearby users. This way, CDN VNFs can be dynamically instantiated based on users' needs, content popularity, viewers' geographical distribution, and mobility patterns. Therefore, in both cases of OSM and Open Baton, we instantiate CDN VNFs as cache servers for a specific website (such as Google Maps), so the users get the website content with an expectedly lower perceptible latency. The motivation to experiment with such type of service is its particular edge-suitability, which means that dynamic instantiation of necessary CDN services significantly affects users' latency [126, 127, 128, 125]. As measuring latency at the user equipment side is out of scope of our paper, we leverage the results provided in [126, 127, 128, 125], which show the latency-related benefits of deploying CDN at the network edge. Thus, the scope of our performance analysis is to measure overall instantiation delay, as the time needed for MANO system to instantiate a network service on top of the MEC platforms.

For the purpose of testing, we created four types of network services, i.e., SFCs. Each SFC consist of one or more VNFs that are chained in order to deliver the full functionality of a final network service. As presented in Table 3.6, SFCs that we created are differentiated by number of VNFs that they contain, i.e., they contain one, two, three, and seven VNFs, respectively. In order to create a fair environment for benchmarking MANO tools, we used the same types of network services for both tools, therefore, customizing VNF and network service descriptors, so they can be interpreted by both orchestrators.

3.3.2 Metrics

In order to assess performance evaluation of Open Baton and OSM, we define the following metrics i) Overall Instantiation Delay (OID) of network service, ii) Overall Termination Delay (OTD) of network services, and iii) CPU and RAM utilization for performing the two aforementioned orchestration operations. As defined in Table 3.7, the instantiation delay is the overall time needed for a network service to be on-boarded and instantiated on top of the NFVI. In order to illustratively explain the aforementioned KPI, we created a sequence diagram (Fig. 3.6), which presents the communication between particular MANO components in the process of instantiating a network service. In addition, Fig. 3.8 illustrates all the processes that contribute to measuring both OID and OTD. OID is a particular metric that can be used to evaluate performance of MANO solutions, based on the time they need to on-board, and to instantiate a network service. On the other hand, OTD is the overall

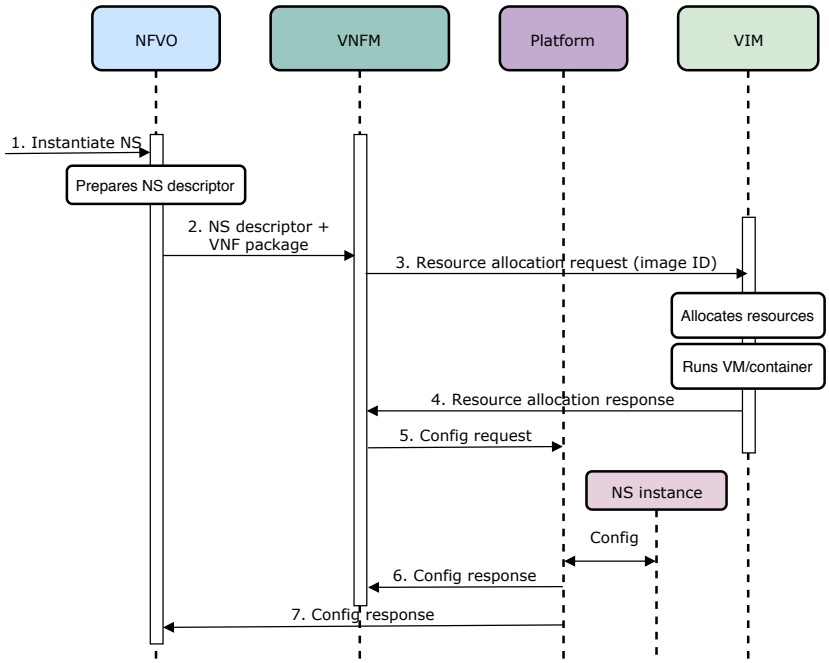


Figure 3.6: The process of instantiation of network service on top of the NFV infrastructure.

time that orchestrators take to terminate the network service instance, and thus, release resources from the MEC platform. Besides OID and OTD, we have also measured CPU, and RAM utilization. In case of CPU, utilization is measured as an average usage of processing resources, i.e., the amount of work with which a MANO solution burdens the CPU of the underlying host. Accordingly, the usage of RAM means the average allocated memory needed for a MANO operation.

Some other metrics are *run-time* metrics that can be used to benchmark the performance of MANO during service execution, when it is up and running (e.g., scaling in and out). The run-time metrics are of high importance for MANO performance, as they directly contribute to perceivable KPIs by users. In particular, when more resources are needed for service operation, orchestrator should re-allocate resources, and scale-up ongoing network service in order to avoid potential disruptions in service operation. However, although stated in their documentation that both MANO solutions support run-time operations, we have revealed that it is not the case. Therefore, benchmarking of MANO solutions is limited on on-boarding and instantiation procedures for now.

3.3.3 The Virtual Wall testbed

For the experimentation setup, we used the Virtual Wall testbed, which is a large-scale generic environment for advanced networking, distributed software, cloud, big data, and scalability research and testing [129]. In overall, the testbed contains more than 400 bare metal and GPU servers which are fully configurable in terms of their software installation,

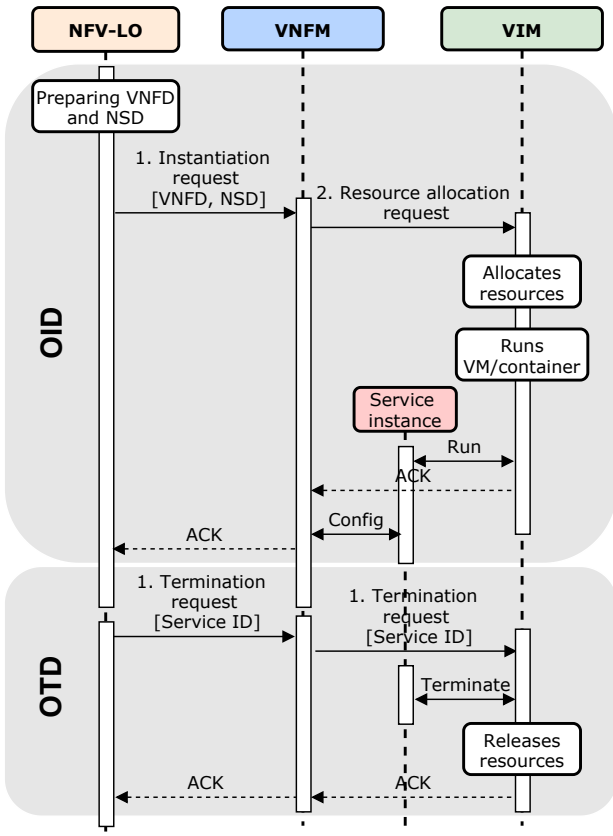


Figure 3.7: Measuring OID and OTD.

as well as the interconnection between network interfaces. Regarding connectivity, each node has a public IPv6 address as well as public IPv4, and thus can be easily accessible from any machine inside or outside of testbed environment. As nodes can be utilized for wide variety of purposes (such as terminal, server, network node, and impairment node), we used three of them to install OpenStack, as virtualization infrastructure, altogether with Open Baton and OSM, as MANO entities [129] (Fig. 3.8 and Table 3.8). The Virtual Wall testbed is a part of FED4FIRE+ [130] project, which is the largest federation of next generation internet testbeds in Europe. Additionally, the testbed is powered by the jFed [131] experimentation toolkit that allows experimenters to push their code to the nodes. It offers to experimenters the possibility of experiment scheduling and a GUI with a real-time information of the experiment execution. jFed platform is supported by Linux Containers (LXC) to submit the code. As shown in Fig. 3.8, we enabled NFV infrastructure resources on top of the testbed infrastructure, in order to be able to instantiate network services.

3.3.4 Comparison of MANO systems: OSM vs. Open Baton

In this section we detail on the performance analysis of two MANO systems, i.e., OSM and Open Baton. In order to approach the experimentation on comparing these two systems,

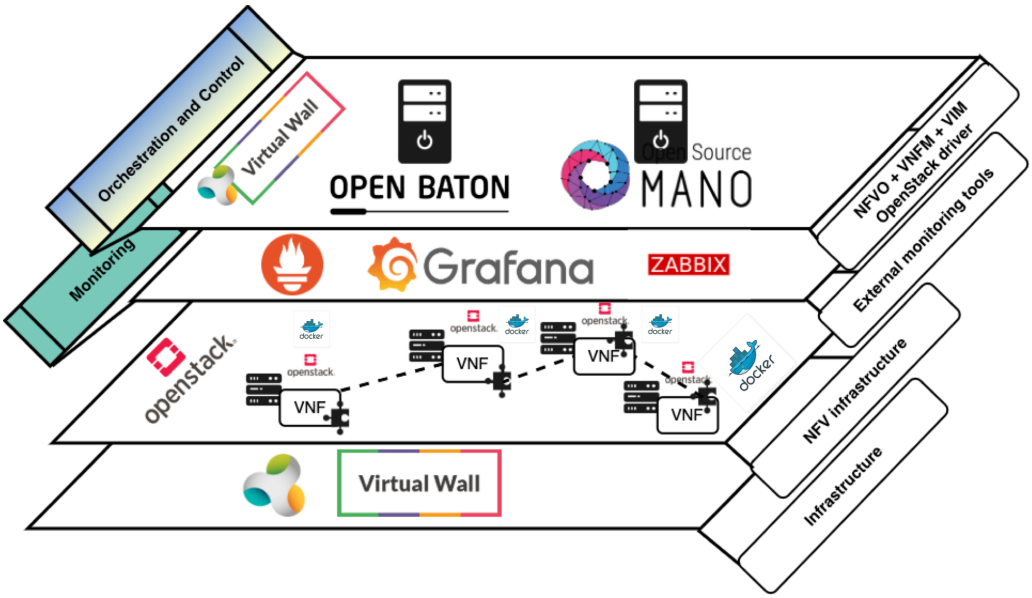


Figure 3.8: Experimentation setup on Virtual Wall testbed.

Table 3.8: Overview of installation within experiment.

Component	Type of machine in Virtual Wall	Capabilities			Operating system
		RAM	CPU	storage	
OpenStack	pcgen4	48 GB	2 x 8 core Intel E5-2650v2 (2.6GHz)	250 GB	Ubuntu 18.04
OSM	pcgen5	16 GB	1 x 4 core E3-1220v3 (3.1GHz)	250 GB	Ubuntu 16.04
Open Baton	pcgen5	16 GB	1 x 4 core E3-1220v3 (3.1GHz)	250 GB	Ubuntu 16.04

Table 3.9: The closed-loop life-cycle management of network services mapped to MANO solutions.

MANO	**MANO components**		
	Orchestration	**Control**	**Monitoring**
Open Baton	NFVO	OpenStack VIM driver	Zabbix plugin
		Generic VNFM	
		Fault management system	
		Auto-scaling engine	
		Network slicing engine	
OSM	Resource orchestrator	OpenStack VIM driver	Performance management
	Service orchestrator	VNFM	
		Fault management	

we created the two following experimental setups:

- *Experiment 1*: we provide a performance analysis of Open Baton and OSM, and compare them based on the overall VM instantiation delay, CPU, and RAM utilization,
- *Experiment 2*: we examine how Open Baton behaves when different virtualization technologies, i.e., Containers, and VMs, are used to instantiate network services.

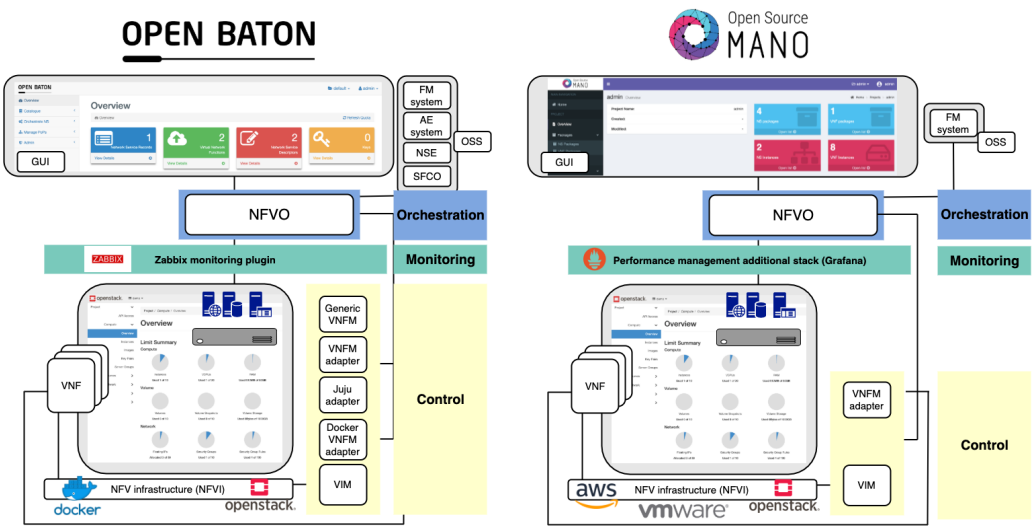


Figure 3.9: Open Baton and OSM architectures mapped to ETSI NFV MANO.

In order to generate *a fair environment* for comparison of Open Baton and OSM, we created Experiment 1 in which we utilized OpenStack as a VIM for both MANO systems. As already shown in Table 3.3, OSM Release Six does not provide support for Containers as a virtualization technology, and therefore, Experiment 2 shows the performance analysis of Open Baton in case it instantiates network services as Containers, and VMs. Regarding the overall experimentation setup, Fig. 3.8 displays the MANO components which were deployed within both of our experiments, altogether with the software components that we used. In particular, the bottom layer is presented as NFV infrastructure, which hosts VNF chains, i.e., network services. As Fig. 3.8 clearly depicts, we used OpenStack, and Docker, to make NFV infrastructure available for instantiating VNFs. Within the middle layer, Prometheus together with Grafana was used as an external monitoring tool for OSM, while Open Baton allowed monitoring via Zabbix external monitoring plugin. Finally, on the upper layer, Open Baton and OSM were installed and set up to embody the roles of orchestration and control.

Being aligned to Fig. 3.4, and the way we mapped particular components of ETSI NFV MANO framework to closed-loop life-cycle management groups (i.e., orchestration, control, and monitoring), the upper layer in Fig. 3.8 comprises both orchestration and control, which means that both processes are performed by MANO entities. Thus, Table 3.9 shows which MANO components belong to particular process.

The middle layer of experimentation setup in Fig. 3.8 is in charge of monitoring tasks, which in collaboration with upper layer, closes the loop of automated life-cycle management of network services. In Table 3.8, specific details on installation of Open Baton, OSM, and OpenStack, are provided.

To realize orchestration of network services and resources, we considered tools with *lighter* installation setup, in order to create a lightweight orchestration environment, suitable for resource constrained MEC platform on the network edge. Due to the capabilities of similar

scale (Tables 3.4 and 3.5), we chose Open Baton and OSM for the experimentation and performance analysis. OpenStack is an open-source software platform for cloud computing, and MEC platform providers consider it as a suitable solution for enabling MEC infrastructure. Following this trend in both industry and academia, we installed OpenStack to provide underlying NFV infrastructure whose resources need to be orchestrated in order to properly host network services. On the other hand, Docker is a platform that enables developing and running the applications, while separating them from the infrastructure, so the software can be delivered quickly [132]. In our case, Docker used resources that were available within the NFV infrastructure on top of which it was installed and configured.

Both MANO solutions are open source platforms with a goal to provide a comprehensive implementation of the ETSI NFV MANO specification for orchestrating heterogeneous NFV infrastructures. Open Baton [133] is built by the Fraunhofer Fokus Institute and the Technical University of Berlin [39]. We installed the latest version which includes OpenStack VIM driver for deploying VNFs on OpenStack, generic VNFM for instantiation of VNFs, Fault Management System (FMS) for detection and recovery of VNF faults, Auto Scaling Engine (ASE) for automatic creation and termination of VNF instances, and Network Slicing Engine (NSE) for ensuring a specific QoS for a network slice (Tables 3.8 and 3.9). OSM [134] is an ETSI-hosted project for delivering open source MANO tool, and the seventh release has been launched recently. Its orchestration functions are divided into two entities: resource and service orchestrator. As presented in [39], OSM integrates several open source software initiatives to deliver fundamental ETSI NFV MANO functionalities. In particular, Riftware is used as a network service orchestrator, OpenMANO as resource orchestrator, and Juju Server as VNFM [39]. We installed OSM Release Six, which enabled the use of service and resource orchestrators, VNFM, OpenStack VIM driver, and fault management (Tables 3.8 and 3.9).

In Table 3.9, we map installed components of both MANO tools to the closed-loop life-cycle management and orchestration. A more illustrative representation of mapping Open Baton and OSM to closed-loop life-cycle management and orchestration, showing their compliance to ETSI NFV MANO framework at the same time, is presented in Fig. 3.9.

3.3.4.1 Results and Discussion

Regarding the overall instantiation time, i.e., OID, Fig. 3.10a shows that performance of both tools highly depends on the number of VNFs chained into network service. In particular, Table 3.6 shows how are particular VNFs (from VNF_1 to VNF_7) connected to the service chains. If we examine the network service complexity, as a number of VNFs that a particular network service chain consists of, we notice the following:

- Open Baton outperforms OSM in case of service function chains with both lower and higher complexity (i.e., lower/higher number of VNFs in SFC). This statement is also supported by a statistical test, i.e., a two-sample t-test[1], that is utilized for inspecting its statistical significance. Thus, we applied the t-test on the collected OID measurements for both Open Baton and OSM, and as a result we obtained

[1]Two-simple t-test is used for evaluating the significance of difference between two populations.

$p_{value} = 0.002192$. For the significance level of 95%, p_{value} larger than 0.05 indicates acceptance of null hypothesis, i.e., the two samples are equal. Therefore, our result shows that the difference between measured OID for Open Baton and OSM is also statistically significant ($p_{value} < 0.05$). The average value of OID for a single VNF is 7.23 s in case of OSM, with standard deviation of 8.45 s, while in case of Open Baton these values are 3.15 s and 2.23 s, respectively. Thus, OSM takes longer to deploy a single VNF on average, while a larger standard deviation among values of OID for VNFs in the service chain indicates a larger variance due to the longer time of deploying a first VNF in the chain. On the other hand, when VMs and containers are compared (Open Baton only), the average time of deploying a VM is 3.15 s, with standard deviation of 2.23 s, while in case of container the average OID and standard deviation are 1.76 s, and 0.83 s, respectively.

- In Fig. 3.11a, and Fig. 3.11b, the increasing trend from SFC_1 to SFC_4 is somewhat expected due the way how SFCs are generated (Table 3.6), i.e., the more VNFs are chained, the more memory and CPU resources are needed for an SFC to properly run. This trend has a lower slope in case of CPU, since CDN services that we instantiated as SFCs do not run CPU-intensive tasks. Furthermore, in CPU (Fig. 3.11b) and RAM (3.11a) utilization results, we did not find significant difference between these two MANO tools, which was expected due to allocating the same flavors of VNF for both tools. In particular, average CPU consumption for OSM is 68.3325% (standard deviation 1.02%), while for Open Baton it is 69.25% (standard deviation 0.58%). The application of a t-test on the samples of CPU measurements for OSM and Open Baton resulted in p_{value} of 0.169, indicating no significant difference between CPU consumption of those two MANO systems (null hypothesis not rejected). The same trend applies to RAM load values, where the t-test for OSM values (average 832.0 MB, standard deviation 673.26 MB), and Open Baton values (average 829.5 MB, standard deviation 673.71 MB) resulted in p_{value} of 0.995 (null hypothesis not rejected2).

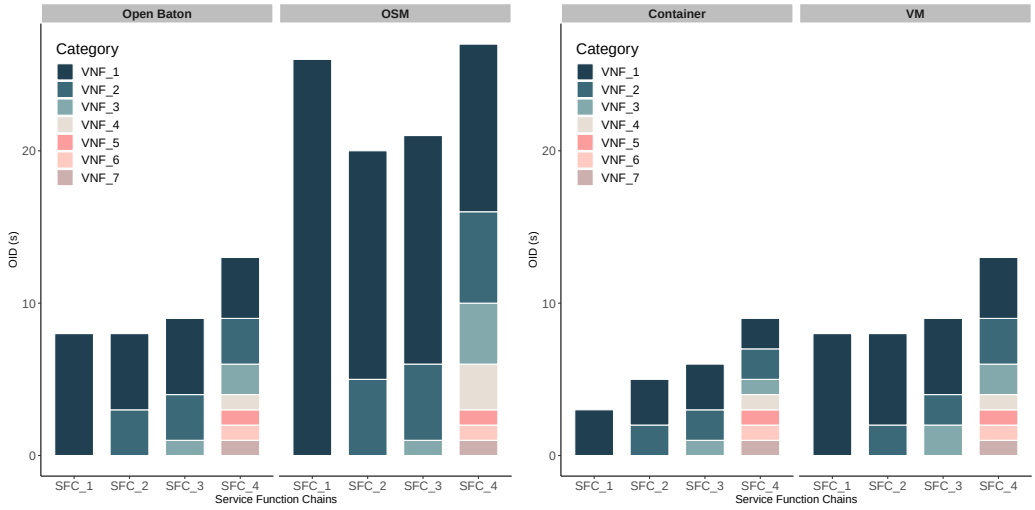
Within confines of the aforementioned observations, we can derive the following conclusions, as perspectives for incorporating Open Baton and OSM into real use-cases of automated closed-loop life-cycle management in MEC-based vehicular networks.

1. Taking into account a feature-based analysis presented in Section 3.2 and Table 3.3, OSM provides a lightweight solution for orchestration of network services and resources, as it requires much lower capabilities than Open Baton. Such advantage makes OSM more suitable for installation and setup on resource constrained edge cloud platform, such as MEC.

2. Regarding compatibility with different VIM environments, the OSM Release 6 supports more VIM drivers than the last version of Open Baton. Thus, the possibilities of customizing OSM to various NFV infrastructure types are broader than in Open Baton.

3. Based on our experience during the experimentation, both tools suffer from insufficient and inconsistent documentation, which complicates the overall process of installation and setting up.

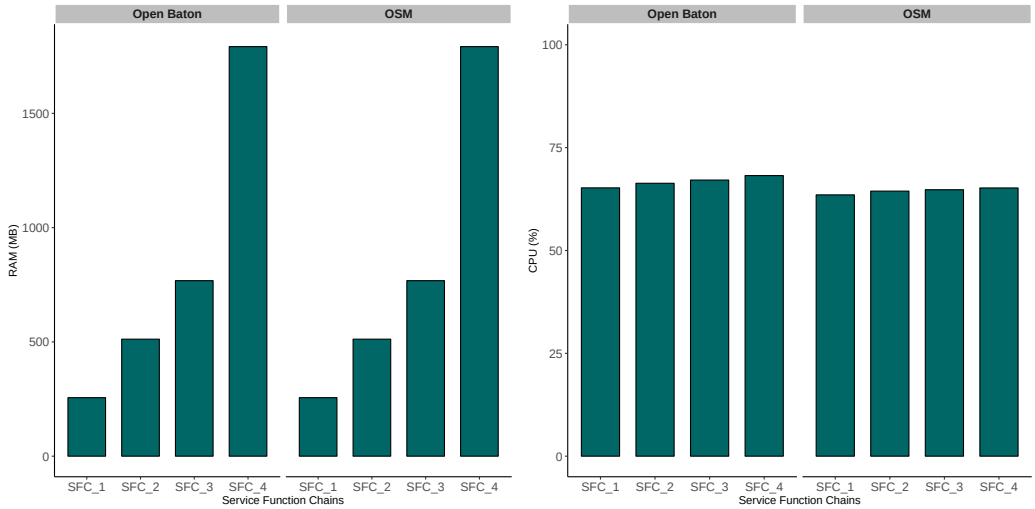
4. As we already emphasized in Section 3.2, the support for container-based virtualization is important if we take into account the limited resource availability in MEC platforms.



(a) Network service instantiation delay: Open Baton vs. OSM.

(b) Network service instantiation delay: Docker Containers vs. VMs.

Figure 3.10: Management and orchestration in MEC-enhanced vehicular networks.



(a) RAM utilization: Open Baton vs. OSM.

(b) CPU utilization: Open Baton vs. OSM.

Figure 3.11: Management and orchestration in MEC-enhanced vehicular networks.

Open Baton supports containerized network services and applications, which is a significant advantage over OSM. Although the latest release of OSM supports Kubernetes [135] as VIM, and accordingly enables containerized service deployment, it is in an early stage, and requires more testing.

5. Aligned to the previous point, results from Experiment 2 shown in Fig. 3.10b show that container-based service instantiation takes less time for each service type, as expected due to the lightweight capabilities of Containers in comparison to VMs. Furthermore,

in order to inspect the statistical significance of our results, we have applied the t-test on the collected measurements for OID. The test resulted in $p_{value} = 0.004332 < 0.05$, which indicates that the difference between OID values for Docker containers and VMs (instantiated upon Open Baton's guidance), is also statistically significant. The difference in overall delay between corresponding container and VM variants are even larger that presented in Fig. 3.10b, because after on-boarding and instantiation procedures, container-based service is ready to be consumed by users, while VMs instantiated on top of OpenStack only got their resources and IP addresses, but the automated configuration of underlying operating system takes 2-3 minutes more.

6. In both Fig. 3.10a, and Fig. 3.10b, we present the values of OID for each SFC as a stacked value, i.e., we show how each of the VNFs (from VNF_1 to VNF_7) contributes to the overall OID, needed for this SFC to be instantiated. In particular, if we take a look at the time needed for SFC_4 to be instantiated, we can see that VNF_1 contributes to the overall OID the most, while the last three VNFs (i.e., VNF_5, VNF_6, and VNF_7) take the least time for their instantiation. It can be depicted in both Fig. 3.10a, and Fig. 3.10b that the impact of the first VNF in the chain on the overall OID is the highest. However, such result is reasonable, and expected, as each of the VNFs are spawned by using the same image, which means that the on-boarding procedure is included in the instantion of VNF_1, and once it is instantiated, all the remaining VNFs will take much less time, since the image is already available to the VIM.

7. From the perspective of overall instantiation delay, we expect that Open Baton will enable more suitable environment for realistic vehicular service implementations, consisted of multiple more or less complex VNFs. As we already elaborated on importance of Ultra-Reliable and Low-latency Communication (URLLC) in automotive use cases, more attention should be paid to prompt service instantiation. However, although lower in case of container-based deployment, instantiation delay for Open Baton is still perceptible, and some pre-emptive methods for predictive instantiation are needed, so the services can be ready on a MEC platform at the moment when they are needed.

8. Taking into consideration all findings based on a realistic example of CDNaaS, none of these two versions of MANO tools are ready to be used in realistic scenarios for vehicular communications, as run-time operations such as service scaling-in and out, muting, migration, etc., are neither mature nor automated.

3.3.5 Comparison of Virtualized Infrastructure Managers (VIMs)

Here we briefly provide an overview of OpenStack[2], AWS[3], and Docker[4], and particular settings that allow them to generate a corresponding VIM environment for MANO systems presented in the previous section.

To run a service on top of OpenStack, the required image should be uploaded via Glance service. The name of the image is then used in VNFD, and NSD, so when a request for

[2]OpenStack documentation: `https://www.openstack.org/`
[3]AWS documentation: `https://docs.aws.amazon.com/`
[4]Docker documentation: `https://docs.docker.com/`

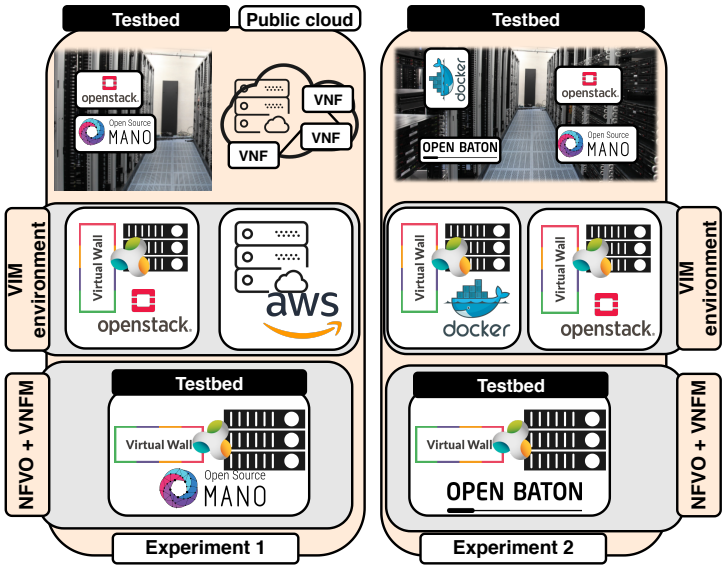


Figure 3.12: Experimentation setup on the Virtual Wall testbed, and the public cloud.

Table 3.10: Supported VIM environments in Open Baton and OSM.

	VIM environment	
Open Baton	OpenStack	Docker
Open Source MANO (OSM)	OpenStack	AWS

instantiation comes from MANO to Openstack, image service retrieves the necessary image for VM instantiation. Furthermore, Nova and Neutron are services that provide compute and network resources based on the flavors and network specifications, that are also stated within VNFD and NSD. In order to register AWS as a VIM for MANO, we needed *access* and *security* keys for our account, flavor of instances that will be instantiated, as well as a corresponding availability zone. Furthermore, to run an instance from MANO, it was necessary to specify a *key pair*, a security group, a subnet, and a location of the image needed for service instantiation. Finally, in our experimentation with Docker as a VIM, for the purpose of running specific network services, we created Docker images instead of creating a VNFD, and then used these custom images to generate NSD, and to launch a network service. Therefore, network service is instantiated as a container on top of the Docker machine in a testbed environment. In order to mimic the realistic features of edge computing, we utilized the *testbed environment* for the case of OpenStack VIM, and Docker VIM, while for AWS VIM we used a *public cloud*.

For the purpose of inspecting the impact of VIM on the performance of MANO systems, we created an experimental setup that is illustrated in Fig. 3.12, including the Virtuall Wall testbed, and a public cloud. We made sure that machines selected for installation of Open Baton, OSM, OpenStack, and Docker, meet their resource requirements. The capabilities of these machines are stated in Table 3.5, same as for the performance analysis described in the previous section. The performance we measured is described as the time needed for a service to be instantiated, and terminated on top of the MEC platforms (i.e., OID, and

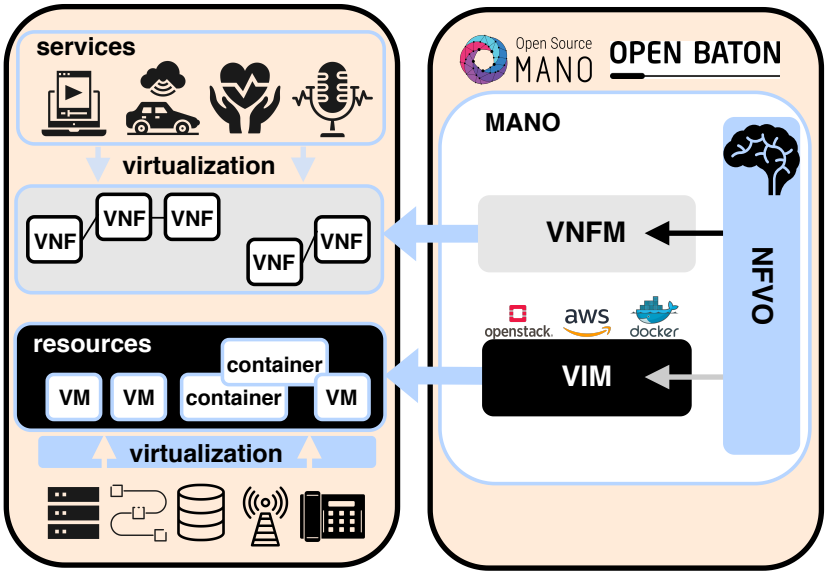


Figure 3.13: ETSI NFV MANO components as a management and orchestration entity for a MEC platform.

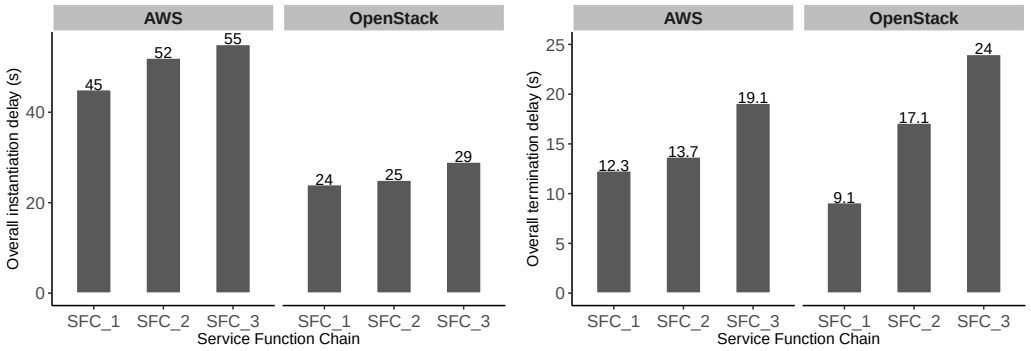
OTD, respectively).The experimentation on evaluating the impact of VIM systems consists of two separate experiments (Fig. 3.12), both measuring the performance evaluation of network service instantiation/termination, as follows:

- *Experiment 1*: setup combining OSM for orchestration (MANO), and OpenStack and AWS for VIMs,
- *Experiment 2*: setup combining Open Baton for orchestration (MANO), and OpenStack and Docker for VIMs.

For both experiments, we tested the performance for three chains of VNFs (SFCs), based on their complexity that is expressed as a number of VNFs contained in the chain (Table 3.6). In Figures 3.14a, 3.14b, 3.15a, and 3.15b, SFCs are: 1) SFC_1 containing one VNF, 2) SFC_2 containing two VNFs, and 3) SFC_3 containing three VNFs.

The testbed configuration of OpenStack mimics the realistic features of edge computing, while for AWS resources, we used the public cloud. Furthermore, in the Experiment 1, in order to create a fair environment for performance evaluation, we instantiated the same types of service (i.e., the same NSD) for both OpenStack and AWS. After instantiation, VMs with Ubuntu operating system (i.e., image uploaded to OpenStack, and present in us-east-1 zone in AWS EC2) were running on top of the OpenStack, and AWS cloud. For the Experiment 2, we measured the OID and OTD of Docker containers that are deployed using the testbed resources, and of VMs of the same functionality, instantiated on top of the OpenStack.

Here we provide a thorough discussion on results shown in Figures 3.14a, 3.14b, 3.15a, and 3.15b:



(a) Experiment 1: OID. *(b) Experiment 1: OTD.*

Figure 3.14: OID and OTD values when measuring the impact of VIM systems on OSM.

1. As shown in Fig. 3.14a, AWS requires more time (s) to instantiate a service, which is on average 50.66 s (standard deviation 5.13) compared to 26 s (standard deviation 2.65 s) in case of OpenStack. The t-test results in p_{value} equal to 0.0017 ($p_{value} < 0.05$), which shows a statistical significance of the difference. This is reasonable since it is a public cloud, and all the internal procedures prior to instantiation are hidden from the user. At the same time, OpenStack provides a dedicated platform (i.e., a private cloud) for user's needs, and it is located at a geographically suitable place for a MANO to orchestrate it.

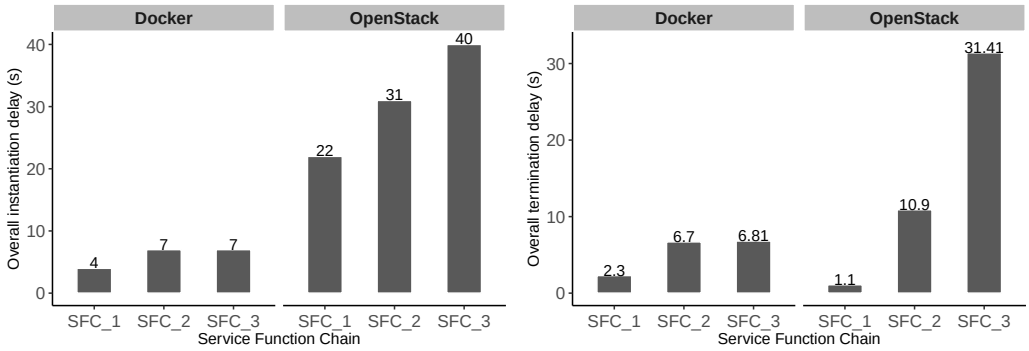
2. Although OpenStack outperforms AWS in terms of OID (Fig. 3.14a), there are configurations and custom installations that need to be done prior to using OpenStack as a VIM, and of course, custom machines are needed (Table 3.5).

3. The more complex the service is, the higher OID and OTD are for all VIMs (Figures 3.14a, 3.14b, 3.15a, and 3.15b). This is somewhat expected, because each VNF, either it is a container or VM-based, takes time for instantiation and termination.

4. Interestingly, AWS needed less time to terminate more complex network services (Fig. 3.14b), i.e., services with two and three VNFs. Thus, once instantiated and went through security procedure, the resources needed for network services can be released in a faster way. On average, AWS takes 15.03 s (standard deviation 3.59 s), while OpenStack takes 16.73 s (standard deviation 7.45 s) to terminate the service. However, the difference is not statistically significant ($p_{value} = 0.74$).

5. Regarding configuration complexity, setting up AWS as a VIM for OSM is not well documented, since additional configurations have to be set-up on AWS as well (security groups, virtual private cloud, and subnets, have to be public in order to communicate with OSM). Such public configuration is not necessary in OpenStack, i.e., networks can be private.

6. Although more variety in flavors and images is present in AWS, there is a certain limitation in creating custom images and flavors based on the users' needs, while in OpenStack this task is straightforward.



(a) Experiment 2: OID. *(b) Experiment 2: OTD.*

Figure 3.15: OID and OTD values when measuring the impact of VIM systems on Open Baton.

3.4 Summary of the Chapter

In this Chapter, we presented a feature-based analysis of the most common MANO solutions and discussed their suitability for orchestrating dynamic vehicular systems. We have carefully selected a set of features that are relevant for orchestrating distributed edge service deployments/EdgeApps, such as: i) resource requirements for MANO system deployment, which are important given that these MANO systems are expected to run at resource constrained edges, ii) support for different virtualization environments, which brings more flexibility in deploying edge services and EdgeApps, given the differences between VM and container-based service deployments, iii) integrated monitoring capabilities, iv) compliance with standards, and v) multi-domain and multi-tenancy support.

Besides analyzing their features, we have evaluated the performance of two of those MANO systems, i.e., Open Baton and OSM, given their similar characteristics in terms of resource requirements and other capabilities presented in the feature analysis. As maintaining required levels of QoS is essential for all latency-sensitive vehicular services/EdgeApps, it is important to understand and evaluate the performance of MANO systems, in order to assess their potential impact on the performance of edge services and EdgeApps. In particular, excessive service instantiation latency is delaying service availability on the network edges, and as such, it could disrupt the vehicular service operation (e.g., notifications on the emergency situation on the road are not distributed, thereby highly affecting the assisted navigation of civilian vehicles). Also, in case of service failures, it is important to re-instantiate the service, and in case of long instantiation delays, the service performance could be severely affected. That is why we focused on measuring overall instantiation delay in case of Open Baton and OSM, to assess their readiness for real-life deployments in dynamic vehicular systems.

From the results we collected during performance evaluation of these two MANO solutions, we learned that OSM offers higher virtualized infrastructure compatibility as it supports more VIMs. However, Open Baton shows an important advantage compared to OSM due to the support to Docker containers, which results in significantly lower instantiation delay compared to VM-based deployments enforced by OSM. Despite this advantage, the instantiation delay in case of Open Baton still reaches the values that are larger than 1 s even

in the case of deploying container-based services and EdgeApps. For instance, if EdgeApp needs to be re-deployed, and the deployment is taking 1 s or longer, this means that users, i.e., vehicles will be without service, which could have an adverse impact on their network assisted navigation. Thus, there is a clear need for studying more pre-emptive methods for instantiation, which will proactively deploy edge services and EdgeApps before the QoS experienced by users is deteriorated.

Such an analysis of both features and performance of existing MANO solutions proved as essential for understanding the design requirements of future MANO systems that need to utilized for orchestrating vehicular edge services and EdgeApps. Thus, the results and insights we obtained from this work were used as a starting point for designing and developing a more comprehensive orchestration framework presented in the following Chapter.

Chapter 4

Collaborative edge orchestration for Connected Cooperative and Automated Mobility

This chapter is part of the **Contribution 2**: *Resource and service orchestration for Connected Cooperative and Automated Mobility*, and it is based on:

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," in *IEEE Transactions on Mobile Computing*, https://doi.org/10.1109/TMC.2021.3118058

N. Slamnik-Kriještorac, G. M. Yilma, F. Zarrar Yousaf, M. Liebsch and J. M. Marquez-Barja, "Multi-domain MEC orchestration platform for enhanced Back Situation Awareness," *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1-2, https://doi.org/0.1109/INFOCOMWKSHPS51825.2021.9484632

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

4.1 Orchestrated and Collaborative Edges as enabler of Secure and Federated CCAM

The 5^{th} generation of the cellular mobile communication system (5G) is being deployed stepwise in the mobile operators' infrastructures, thereby promising low-latency and high bandwidth communication services to not only mobile devices but also to vertical industries

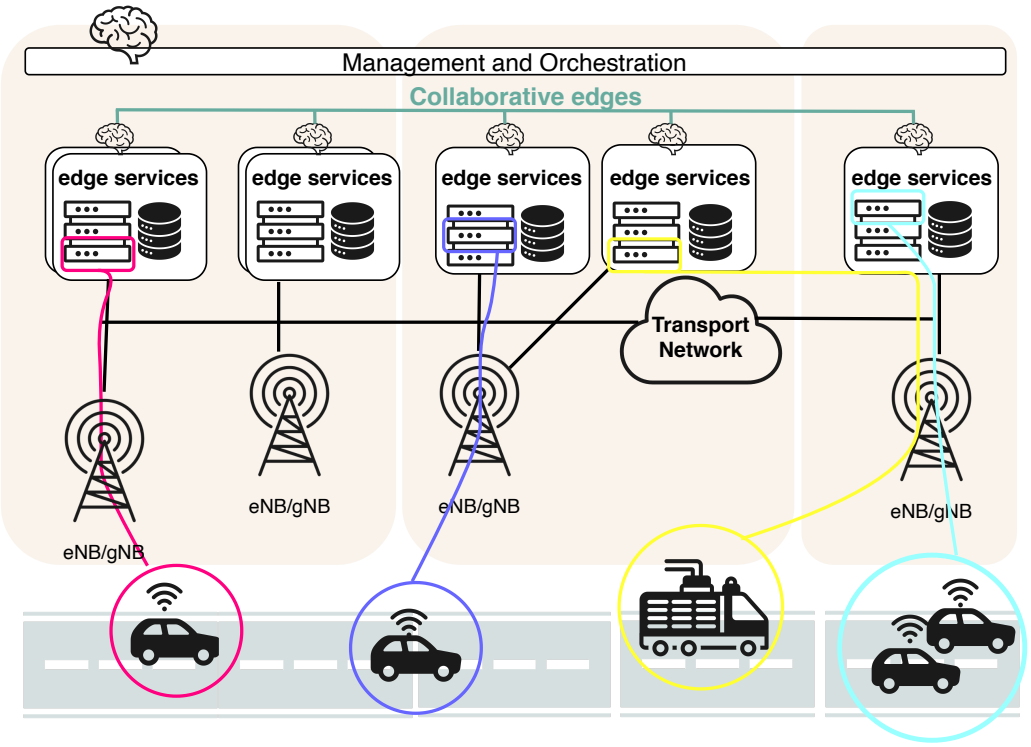


Figure 4.1: High-level overview of collaboration between orchestrated network edges that host edge services for vehicles.

with diverse service requirements in a resource and energy efficient manner. The NFV, being one of the main technology enablers of 5G, affords the 5G core network architecture to follow a clear control-/data plane separation. This separation enables automated and agile deployment and Life-cycle Management (LCM) of the associated VNFs, constituting to deliver customized network services catering to a variety of use cases over the same 5G network infrastructure. Furthermore, MEC systems are being widely deployed in the edge networks to deliver a low-latency and localized access to virtualized services by deploying them in close proximity to the users. However, due to the fact that 5G Core, NFV and MEC technologies are being developed by different standardization bodies, the deployment, integration and inter-play of these solutions in support of the expected features and end-to-end performance figures of such 5G ecosystem is not coordinated.

The challenge is thus to develop an integrated framework for the automated deployment and orchestration of an end-to-end network in support of the expected service quality. Such framework should span i) the provisioning of virtualized service instances in a centralized cloud, ii) the configuration of a transport network, which connects the service cloud with the cellular network of a mobile network operator, and iii) the configuration of the mobile radio access. The resulting architecture enables full control of the network in between centrally deployed services, and mobile devices, which connect to these services through the cellular 5G network. The automotive industry represents a promising yet challenging

consumer of such 5G ecosystem that has the potential of enabling novel and performance critical use cases that were not possible with the previous generations of mobile network systems. This is especially true in the domain of assisted and autonomous driving that primarily relies on real-time and enhanced situation awareness involving high-density, low-latency, and complex processing, of the vehicular sensor data. This entails for consistent quality and low-latency communication with infrastructure service functions. As MEC systems enable low-latency due to exposing resources to the network edge, decentralization and distribution of the virtualized service functions towards the cellular network edge help to deploy services topologically closer to vehicles (as depicted in Fig. 4.1). Such deployment enables the collection, processing, and provisioning, of data locally where they are generated and needed, at the same time shortening the communication path and contributing to a reduced latency as well as to core network traffic offload. However, in such highly agile automotive environment, service continuity in low latency communication with distributed services at the cellular network edge requires real-time monitoring and seamless reconfiguration as well as relocation of the connection to a service instance closer to the vehicle. To enable service continuity and promised KPIs (e.g., high reliability, low latency, and high throughput), management and orchestration systems need to be effective to provide distributed service deployment, and seamless service reconfiguration and relocation in such highly mobile and resource constrained ecosystem. Reactive approaches for service continuity, which adjust a configuration after an event happened, such as a vehicle moving to a location which can be served by a closer network edge, are more and more complemented or even replaced by proactive solutions, which leverage data analytics, machine learning, and artificial intelligence for the anticipation of such event and the in-advance preparation of the network.

Despite the low latency benefits for CCAM services enabled by deploying services close to the vehicles, MEC deployments pose acute challenges in terms of the management and orchestration of virtualized services in a resource constrained and highly distributed environment, which if not properly managed can have adverse impact on the end-to-end service latency and service reliability. This is because of the distributed nature of the multi-domain MEC environment, where even a single domain (e.g., PoP) may have multiple geographically dispersed MEC sites. Each MEC site offers an NFVI with limited compute/network/storage resource footprint (i.e., MEC host), managed by a local platform manager/orchestrator. To manage the distributed service deployments across MEC sites i) a coordination between the respective platform managers/orchestrators is required, with an additional coordination in case the service deployment encompasses MEC sites belonging to different administrative domains (e.g., countries), and ii) as service deployments may belong to different tenants, strict isolation between service instances need to be ensured without compromising QoS. The aforementioned challenges can be mitigated by enabling collaboration between orchestrated edges via the hierarchical distribution of orchestration tasks, which provides proactive multi-domain service deployments with support for service continuity. Thus, in this Chapter, we propose and investigate in detail an architecture of a multi-tier orchestration platform for CCAM, and associated operations in support of orchestrated distributed mobile edge networks, in order to enable service continuity for vehicles, which connect to distributed mobile edge services (see Fig. 4.1). The presented solution extends prior work [136, 137] on the end-to-end orchestration in the autonomous operation of orchestration tasks at mobile edge network as well as the connectivity between edge orchestration functions of geographically and topologically adjacent mobile edge networks, aiming at optimized edge-to-edge service continuity by enabling collaboration between mobile edge networks. The proposed orches-

tration platform aligns with specifications of relevant standardization bodies (i.e., 3GPP, ETSI MEC, and ETSI NFV) and builds on top of the 5G System specification, which, when compared to previous generations of the mobile communication system, provides various advantages at architectural, protocol, and operational levels. This includes i) the support of a decentralized data plane and edge computing by means of the already mentioned clean control/data plane separation, and ii) the adoption of service-based communication principles and the use of web communication protocols (such as REST, and Google Remote Procedure Calls (gRPC)) at the 5G control plane, which eases the integration of and interworking with control and management functions of accompanying systems, such as edge computing systems and orchestration systems. Specifying the 5G architecture as a set of service producer and service consumer functions, which apply service-based communication, matches a cloud-native design and suits a deployment on top of an NFV infrastructure with automated management and orchestration, as described in this article, with the focus on distributed edge clouds. The promised benefits of 5G system in terms of e.g., the ultra-low latency and high bandwidth depend on the efficiency of the management and orchestration of resources and service, as if there is no collaboration between distributed edge clouds established by orchestration layers, service performance and service continuity will be affected, thereby leading to service performance degradation. The flexible deployment and use of the 5G System's data plane functions and the specified support for Service and Session Continuity (SSC) [138], which permits changes and adjustments in the data plane configuration without disrupting the mobile data session, enables local breakout of mobile data plane traffic and maintains access to edge computing resources and hosted edge services (Fig. 4.1). The presented solution provides new extensions for MEC-5G System coupling, management and orchestration reference points between mobile edge network orchestration functions, as well as for automated local orchestration at and between edge networks per customized policy for autonomous orchestration tasks, denoted as Management Level Agreements (MLAs) [139].

The analytical and experimental evaluation of the performance of collaborative orchestration is presented to substantiate the design choices that are made to tackle highly mobile use cases with intrinsically distributed service deployments. The evaluation is based on the KPIs associated with a deployment per the proposed architecture. These KPIs are i) the average response time needed for performing orchestration operations, ii) the load of the orchestration entities that needs to be balanced across distributed and multi-layered orchestration systems, and iii) the average power consumption of the performed orchestration requests. In the context of the aforementioned KPIs, it is of utmost importance to assess the load that any orchestration entity is exposed to, making sure that these entities can handle all the orchestration requests in a required response time frame. In particular, with the analytical and experimental evaluation of the orchestration platform, we aim to achieve the following goals:

- To determine how the number of available instances of reference points in the orchestration platform impacts the communication delay in the average response time of an orchestration request, as well as the amount of resources available for performing this request.
- To assess how the number of available instances of reference points between the distributed orchestration components affects the load of the orchestrators at different

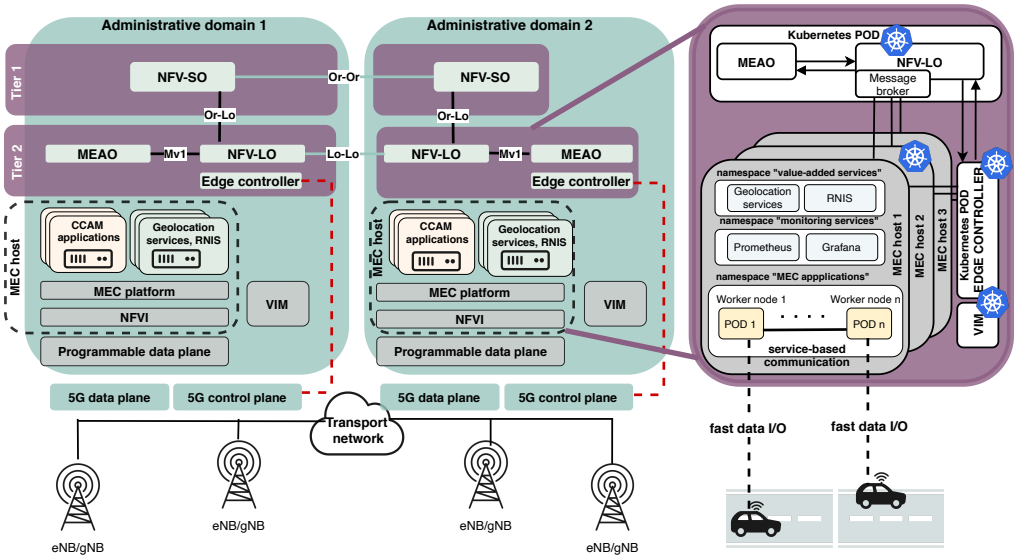


Figure 4.2: High-level functional architecture of the orchestrated platform for CCAM in a federated configuration.

hierarchical tiers, and how this load further impacts the average response time.

- To showcase the benefits of direct interfaces between orchestrators.
- To determine the impact of orchestration operations on the overall power consumption in the system.
- To showcase how the orchestration operations affect service continuity.

In the rest of the Section 4.1 we present how these goals are achieved. In particular, Section 4.1.1 presents the functional overview of the orchestration platform for CCAM among collaborative edges, followed by the key design features (Section 4.1.2), operational aspects (Section 4.1.3), and software design principles (Section 4.1.4). Furthermore, in Sections 4.1.5 and 4.1.8, the analytical model, and experimental evaluation of the orchestration platform for collaborative edges, are presented respectively.

4.1.1 Functional Overview

In this section we present the functional overview of the orchestration platform for CCAM among collaborative edges, its design features, operational aspects, and the software design principles. A platform prototype is being deployed in an automotive-related pilot of the 5G-CARMEN project[1], on top of the MNOs' NFV and wireless network infrastructure. In Fig. 4.2, we illustrate the high-level functional architecture of the orchestration platform for CCAM in a federated configuration, indicating the main components that enable secure and federated cross-domain management and orchestration of 5G collaborative edges.

[1]5G-CARMEN: `https://5gcarmen.eu/`

The orchestration platform for CCAM is designed following the cloud native principles while being aligned with the standardization framework provided by ETSI MEC [50], ETSI NFV [140], and 3GPP [141, 138]. This design enables collaboration between 5G edges, thereby extending the range of the services/applications running on top of these edges, and allowing them to collaborate with peering service/application instances in different domains in order to enable service continuity.

As illustrated in Fig. 4.2, the MANO tasks, such as service on-boarding, instantiation, scaling, migration, and termination (more details provided in Section 4.1.3), are performed by hierarchically organized orchestration platform elements that are distributed in two following tiers [142]: i) top-level service orchestration, and ii) edge-level service orchestration. Such functional split enables offloading, or delegating, the orchestration tasks from top-level orchestrator to the edge-level orchestrators in order to decrease the processing load at the top-level orchestrator while enabling low-latency MANO operations directly at the network edges. The top-level orchestrator, characterized by the NFV Service Orchestrators (NFV-SOs), is a centralized service orchestrator that represents larger network domains on the MNO level. On the other hand, the distributed edge-level orchestrators, characterized by a combination of NFV Local Orchestrator (NFV-LO), MEAO, and Edge Controller, are in charge of particular edge domains, within a larger MNO domain, in which the virtualized functions/applications are running. There is a 1:N relationship between the NFV-SO and NFV-LO/MEAO, while there is further a 1:M relationship between the NFV-LO/MEAO and Edge Controller ($N, M \in \mathcal{N}$).

The orchestrators interface with each other, and federate with their peer orchestrators in another MNO domain over well-defined reference points. Following are the three main reference points: i) the Or-Or reference point, which is based on the ETSI NFV standard [140], and is responsible for federating between the NFV-SOs in different administrative domains, ii) the Lo-Lo reference point, which is derived from the Or-Or reference point, and enables the coordination between the NFV-LOs for supporting state migration, service continuity, and low-latency service orchestration requirements, and iii) the Or-Lo reference point for coordinating the orchestration tasks between NFV-SO and NFV-LO. The interfaces on these reference points inherit from the standard ETSI NFV/MEC reference point interfaces with relevant extensions, such as Lo-Lo and Or-Lo as described above.

Within a single edge domain, the NFV-LO and MEAO coordinate the LCM of virtualized applications related to low-latency and mission critical services that are deployed in MEC platforms at same or different MEC-sites within an MNO domain. These applications consume MEC Value-added Services (VASs) (e.g., geolocation services, and Radio Network Information Service (RNIS)) to enhance their operation. Each MEC platform, which offers an NFVI, is managed by an Edge Controller which, according to ETSI MEC [50], is in charge of MEC Platform Management, and enforces orchestration and LCM operations as per the directives of the orchestration tiers (i.e., NFV-LO/MEAO). The Edge Controller also supports coupling with the 5G mobile network infrastructure for alignment of connectivity to edge services with device mobility.

4.1.2 Key Design Features

Aiming at orchestrated mobile edge networks within a 5G ecosystem, we define and comply with the following key design features:

Coupling of 5G and MEC/NFV In the view of an intrinsically sound 5G ecosystem, the so far separately treated specifications for a 5G System, MEC, and NFV MANO, need to interface and interact for complete end-to-end system management and control. This is to ensure alignment of policies and configurations associated with a mobile subscriber and its data plane on the one hand side, but to keep a certain level of independence between the two systems for the decision and enforcement of local policies. For this purpose, an Application Function (AF) per the 3GPP architecture specification [141] is co-located with the Edge Controller to connect to the 5G System's Control Plane through service-based communication per the 5G architecture's *Naf* reference point. This reference point enables the retrieval of a mobile subscriber's data plane configuration and to subscribe to events in the 5G Control Plane for receiving event notifications, e.g., from the 5G Session Management Function (SMF) after a change in a mobile subscriber's UPF per SSC mode 3 operations during mobility. The Edge Controller holds the control function of a programmable data plane to enforce traffic treatment rules in alignment with the 5G data plane and to enable, for example, metering and traffic steering within the MEC System's network domain, e.g., for load balancing, failover handling or traffic forwarding towards a different MEC Platform or MEC System.

End-to-end mobile data plane control Complementary to the previously described design feature, this feature leverages the MEC System's awareness of a mobile subscriber's data plane policy and configurations to enforce aligned traffic treatment rules in between the UPF and the MEC service. This feature builds on top of the 5G System SSC mode 3, which enables mid-session relocation of a mobile subscriber's UPF without breaking the Packet Data Network (PDN) session by a MEC System that is able to follow a relocated UPF of a mobile subscriber connected to a MEC service. Meeting this design feature enables the maintenance of an optimized routing path between a mobile subscriber and its device, i.e., the vehicle, and the mobile network edge service to which it connects. A resulting continuity in a service with short communication paths contributes to the raised low-latency requirement.

Delegation of MANO operations in a federated environment In order to optimize the performance of the MANO operations, one of the design features is the introduction of the concept of MLA [139], which allows for the delegation of MANO tasks/operations between the top-level and edge-level orchestration systems, and also between the peering edge platforms in same and/or different domains. The MLA is negotiated over the Or-Lo reference point between the two tiers within the MNO domain. The MLA also governs the coordination between the peering NFV-LOs over the Lo-Lo reference point. MLA enables the offloading of LCM operations from the top-level to the edge-level orchestrators. Such negotiated agreement determines the operations and functions that the edge-level orchestration

entities are allowed to perform within their edge boundaries, thereby executing LCM operation on the relevant service applications and their respective resources [142]. Moreover, the prerequisite for establishing cross-domain federation interface, such as Or-Or and Lo-Lo, is an MLA negotiated between administrative domains, i.e., relevant NFV-SOs. Developing federation over Lo-Lo enables the inter-working of edge/MEC and associated edge/MEC platforms, in order to provide a cross-edge on-demand management and orchestration in a collaborative manner, while enabling and maintaining low-latency edge-to-edge CCAM service/session continuity and seamless state migration of users.

Application-specific support for orchestration operations The edge-level orchestrators constantly monitor the deployed edge services, i.e., edge applications, and allow these application instances to send notifications, as well as triggers for certain orchestration operations. To facilitate and enhance the orchestration operations (e.g., proactive service instantiation, and service migration), the application itself can proactively send notifications to orchestration entities. These notifications may reflect some application-specific data, e.g., retrieved from the data plane packets from users, which are not known by orchestrators. The orchestration entities receive such notifications (e.g., by subscribing to the notification topics with pub/sub, or by receiving them on-demand with request/response), and map them to the policies and necessary orchestration operations. For vehicular applications, such notification might signal that a vehicle is moving out of the range of a specific MEC host, and that proactive deployment of another application instance, including service migration, will be needed. Thus, this feature is significantly important for our platform as it can leverage the applications for receiving additional information and event notifications in support of orchestration tasks, i.e., to trigger suitable orchestration operations that will enhance the support for service continuity.

4.1.3 Operational Aspects of the Orchestrated Platform

As outlined in Section 4.1.2, our orchestration platform supports cross-edge/cross-domain management and orchestration, and thus, NFV/MANO operations that are standardized by ETSI [140, 50] need to be optimized to support multi-domain/cross-edge operation. The baseline set of NFV/MANO operations, which our orchestration platform for CCAM supports, consists of: i) application on-boarding, ii) application instantiation, iii) application scaling, iv) application state migration, and v) application termination. Our platform extends beyond these baseline operations to additionally support and enable a) multi-edge service deployment, and to maintain b) edge-to-edge service continuity, the process of which is summarized below with reference to Fig. 4.3. Listing 1 describes high-level steps of multi-edge service deployment operation, and maintaining edge-to-edge service continuity, presented in Fig. 4.3 (steps 1-19).

4.1.3.1 Multi-edge service deployment

The operation is depicted in the Phase 1 of Fig. 4.3. It starts with the top-level orchestrators (i.e., NFV-SO) selecting the edge-level orchestrators (i.e., NFV-LO) that is most appropriate

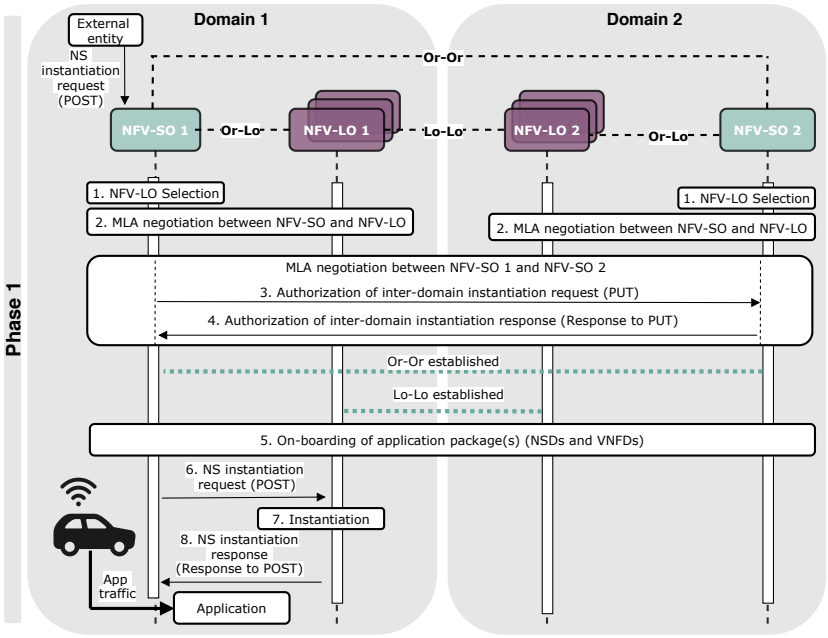
```
Steps 1-4: This operation assumes that NFV-SOs a priori advertise their respective NFV-LOs
↪ and establish MLA via Or-Or interface on a set of orchestration operations to be
↪ delegated to the respective NFV-LOs to collaborate via Lo-Lo interface.
Step 5: On-board network service packages, i.e., VNFDs and NSDs in participating MEC
↪ domains
Steps 6-8: Deploy first instance in domain 1 and perform LCM operations as required
Steps 9-12: While user is about to move to the next domain deploy instance in the new
↪ domain
Step 13: Perform data sharing between two peering application instances
Steps 14-17: Migrate important user state information to next instance to take over the
↪ service, and seamlessly relocate the service endpoint of the user to the new instance
Step 18: Terminate instances that are not in use any longer
Step 19: Notify respective NFV-SOs about termination
Step 20: Repeat steps 9 to 19 as required
Step 21: End
```

Listing 1: Proactive deployment of peering services, and maintaining service continuity in a multi-domain MEC system.

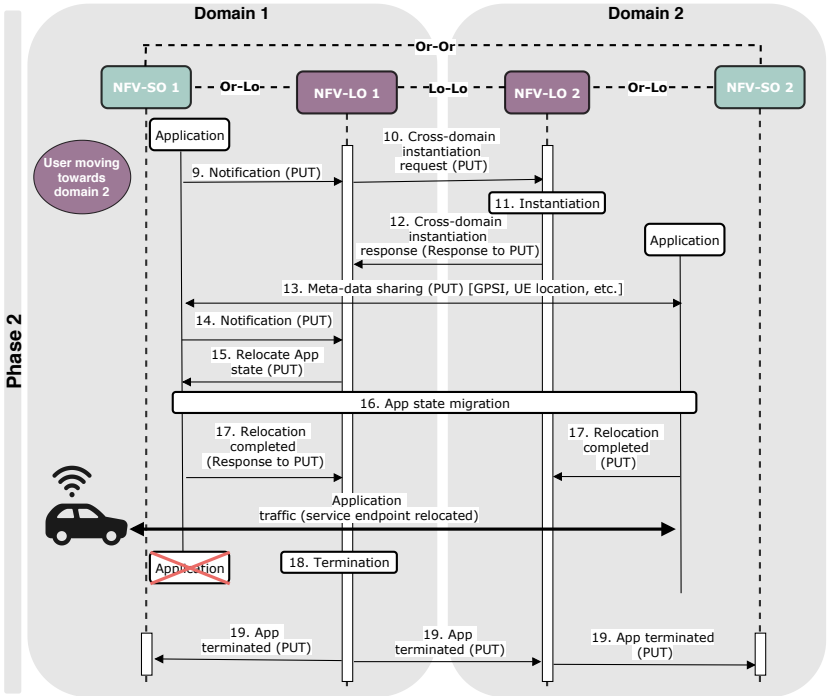
to the service needs (see step 1 in Fig. 4.3). Note that the NFV-LO selection process is out of scope of this section. An MLA is negotiated between the NFV-SOs and the selected NFV-LOs within the respective domains in order to grant management autonomy to the NFV-LOs (see step 2 in Fig. 4.3). For inter-domain operation, a federation is established between the two domains characterized by the establishment of the Or-Or and Lo-Lo reference points between the NFV-SOs and the NFV-LOs respectively [143]. Moreover, the MLAs are also negotiated between the federating NFV-SOs over the Or-Or reference point in order to inform, determine, and harmonize, the scope of management autonomy required between the peering NFV-LOs in order to directly exercise granted LCM operations on the multi-domain deployed application instance over the Lo-Lo reference point (see steps 3 and 4 in Fig. 4.3). Prior to the application instantiation, the orchestration platform performs application package (i.e., VNFDs and NSDs) on-boarding as per ETSI NFV rules (step 5 in Fig. 4.3). In a multi-domain service deployment scenario, the application package can also be proactively on-boarded in the selected peering domains if an MLA exists between these selected platform domains. Afterwards, the NFV-SO will send a service instantiation request, triggered by an authorized external client (e.g., traffic management authority), to the NFV-LO and the service is instantiated (steps 6-8 in Fig. 4.3). Based on the change in user's location, which is being tracked by the application instance, the NFV-LO will receive notification from the application about the need of a peering application instance in the target domain (step 9). This will prompt the NFV-LO 1 to trigger NFV-LO 2 over the Lo-Lo reference point to instantiate the peering application instance in its domain (see steps 10-12 in Fig. 4.3) while the vehicle is still in domain 1. Thus, such proactive instantiation of service in the target domain by direct interaction between the peering NFV-LOs and bypassing the NFV-SOs decreases latency in orchestration operation execution.

4.1.3.2 Edge-to-edge service continuity

To reach QoS levels promised by 5G in terms of ultra-low latency (of 1 ms-10 ms), high capacity (above 100 Mbps per user), and reliability (99.999% availability) [144], it becomes



(a) Proactive deployment of peering services (Phase 1).



(b) Maintaining service continuity (Phase 2).

Figure 4.3: Message sequence chart of orchestration operations in the orchestrated multi-domain MEC system.

imperative for the network service management systems to follow the user mobility, and to place network services always at the most suitable MEC platforms (e.g., the closest one) [110, 111], while maintaining edge-to-edge service continuity. In this context, having in place efficient means for service migration and data plane steering is a challenging proposition where service/application instances or users' session states of ongoing services are relocated from one edge to another as the user moves. Since network edges are usually resource constrained (both network and computing), migrating the application or a user's state needs to be network and resource aware and thoroughly orchestrated. To enforce a smooth service relocation strategy, our orchestration platform enables meta-data and state-data sharing between the multi-edge deployed service instances. This enables application instances to share meta-data (step 13), and to transfer application state (e.g., security token) in case of stateful applications (see steps 14-17 in Fig. 4.3), before a user/vehicle reconnects from source to target instance. The shared meta-data can include the information about the general context of the mobile user/vehicle (i.e., parameters of users' context/session state), such as user's location [106], or Generic Public Subscription Identifier (GPSI) as an identifier in 3GPP, which can further share this data with the target instance, thereby enabling a smooth re-connection of user from one application instance to another. The communication between service instances themselves, and between service instances and vehicles, is accomplished by two types of communication principles, i.e., i) through service based communication leveraging service communication proxies, e.g., to transfer users' session state information to peering instances in adjacent edges, and ii) through fast data I/O interfaces and a programmable data plane to steer and forward data plane traffic to a new location for seamless service continuity (Fig. 4.2).

4.1.4 Software design principles of the orchestration platform

As mentioned above, the design of the orchestration platform for CCAM follows the cloud native principles, which means that all functional elements are implemented as container-based pieces of software rendering a highly modular design. The modularity enables a mix and match of different open source software solutions (e.g., NFV-SO is based on existing OSM). The interfaces between orchestration components (i.e., Or-Or, Lo-Lo, Or-Lo, Mv1, and NFV-LO-Edge Controller, as presented in Fig. 4.2) are implemented following the service based architecture. These interfaces use REST based communication.

For the purpose of developing architecture elements, we use the K8s[2] platform. As depicted in Figure 4.2, the MEAO/NFV-LO are implemented as separate containers within a K8s Pod[3], thereby managing the MEC applications and services via a message broker. Similarly, the MEC applications and services are implemented as container applications in different K8s Pods within each MEC host. The on-boarding procedure, described in Section 4.1.3, practically entails the preparation of Docker images for the MEC applications and services on all required edges. Each Pod with an instance of a CCAM service application can be equipped with one or multiple customized network interfaces, such as for service based communication and data sharing with other application instances, or for fast data plane I/O and associated low-latency communication with other application instances or service clients, as described in

[2]Kubernetes: `https://kubernetes.io/`

[3]Kubernetes Pod is the smallest deployable unit of computing that can be created and managed in K8s.

Section 4.1.3.2. These MEC applications and services are grouped in different namespaces to ensure isolation for performance reasons. Moreover, a monitoring service comprising Prometheus and Grafana are configured in a separate monitoring namespace for collecting real-time metrics and usage statistics for all MEC hosts belonging to the edge domain and to be consumed by the orchestration entities. For the management and orchestration of the MEC applications/service an Edge Controller is configured a separate namespace running as K8s Pod.

4.1.5 Analytical model of resource management and orchestration operations

In this section we provide the analytical model of resource management in multi-tier hierarchical orchestration platforms that are designed for the 5G ecosystems. We first present the resource assignment problem for the distributed service deployments across network edges, and then provide the latency performance analysis for the orchestration tasks performed by orchestration entities in different tiers. Such analytical approach followed by experimental assessment in Section 4.1.8 can be applied to different orchestrated edge solutions, and here we substantiate our design choices, defined for highly mobile use cases with distributed service deployments.

In particular, the impact of the number of available instances of reference points in a collaborative orchestration platform on latency of an orchestration operation, and on a number of hops for an orchestration request, is further studied and presented in Section 4.1.8, by analyzing the response time and the load of different orchestration tiers. Thus, in Section 4.1.8 we analyze KPIs in a greater detail, while in Section 4.1.8.6 we discuss both the analytical model and the results obtained in experimental assessment. As introduced in Section I, the main evaluation goals that we target to achieve with the analytical evaluation in this, and experimental evaluation in the next section, are summarized as follows:

- To determine the impact of the number of available instances of reference point in the orchestration platform on the average response time of an orchestration request, as well as on the amount of resources available for performing this request.
- To assess how the number of available instances of reference points affects the load of the top-level orchestrator, and how this load further impacts the average response time.
- To show the benefits of direct links between edge-level orchestrators.
- To test the power efficiency of the orchestration platform.
- To study how the orchestration operations affect service continuity.

4.1.6 Resource assignment problem

The analytical model of our collaborative orchestration platform defines the resource assignment problem as an integer program. In the Table 4.1, we present the parameters that are

Table 4.1: Parameters in the resource management model.

Parameter	Description
Resource assignment problem	
s	top-level service orchestrator (NFV-SO)
l	edge-level orchestrator (NFV-LO)
i	application implementation
n	MEC host/node
$N_L(s)$	number of NFV-LOs in the domain of NFV-SO s
r	resource
k	type of resource
ρ_{nk}	amount of resources of type k that are available on the n-th MEC host
c_i	cost vector for application implementation i
c_{ik}	cost of resources of type k needed for application implementation i
$d(l)$	administrative domain of l-th NFV-LO
x_{sl}	indicates the relation between s-th NFV-SO and l-th NFV-LO
x_{slin}	decision variable that indicates the ability of l-th NFV-LO to perform orchestration operations on the application implementation i, which is hosted on the n-th node in s-th NFV-SO's domain
Latency performance	
a_{orch}	request for orchestration operation
N_{orch}	number of different orchestration operations
$f(a_{orch})$	traffic generated by orchestration operation request a_{orch}
t	unit time-slot for transmission of an orchestration request via network link
α_{l_1,l_2}	overall transport network latency
$\alpha_{t_{l_1,l_2}}$	transmission delay
$\alpha_{p_{l_1,l_2}}$	propagation delay
$\alpha_{c_{l_1,l_2}}$	computing delay
$\alpha_{q_{l_1,l_2}}$	queuing delay
β, γ	weighting factors that balance network characteristics
$l_{i,j}^{(l_1,l_2)}$	length of the link segment (i,j) that is chained to form the overall link between local orchestrators l_1 and l_2
$B_{i,j}^{(l_1,l_2)}$	bandwidth of the link segment (i,j)
s	speed of electromagnetic signals

utilized in the analytical model. The resource assignment problem refers to the resources that can be assigned to edge-level/local orchestrators, i.e., NFV-LOs, in order to perform orchestration operations for the requested MEC applications.

Table 4.2: Sets of elements in the resource management model.

Parameter	Description
N_S	number of NFV-SOs, $N_S \in \mathcal{N}$
N_L	number of NFV-LOs, $N_L \in \mathcal{N}$
N_I	number of implementations, $N_I \in \mathcal{N}$
N_H	number of MEC hosts/nodes, $N_H \in \mathcal{N}$
N_K	number of resource types, $N_K \in \mathcal{N}$
$\mathcal{S}$	set of NFV-SOs ($s \in \mathcal{S}$, $\mathcal{S} = \{1, \ldots, N_S\}$)
$\mathcal{L}$	set of NFV-LOs ($l \in \mathcal{L}$, $\mathcal{L} = \{1, \ldots, N_L\}$)
$\mathcal{I}$	set of implementations ($i \in \mathcal{I}$, $\mathcal{I} = \{1, \ldots, N_I\}$)
$\mathcal{H}$	set of MEC hosts ($n \in \mathcal{H}$, $\mathcal{H} = \{1, \ldots, N_H\}$)
$\mathcal{R}$	set of resource types ($k \in \mathcal{R}$, $\mathcal{R} = \{1, \ldots, N_K\}$)

Table 4.3: Scenarios for calculating the total number of reference points.

Scenario	Number of NFV-SOs	Number of NFV-LOs in NFV-SO 1 domain	Number of NFV-LOs in NFV-SO 2 domain
I	2	2	1
II	2	1	1
III	2	3	2

In this analytical model, we consider the orchestration platform for CCAM as a hierarchical NFV management and orchestration architecture that consists of the top-level, and the edge-level orchestrators, i.e., NFV-SOs and NFV-LOs, respectively, and as described in Section 4.1, we consider three types of reference points that connect them, i.e., Or-Or, Lo-Lo, and Or-Lo. The sets of elements used in our analytical model are shown in Table 4.2.

Depending on the MLAs that are agreed between NFV-SOs and NFV-LOs in all edge and administrative domains, there is a different number of interfaces that are established on-demand between different orchestration entities. Therefore, the equation (4.1) represents the total number of interfaces that are established *on-demand* between: i) all existing NFV-LOs and NFV-SOs (Or-Lo), enabled by MLA type m_1, ii) all existing NFV-SOs between themselves (Or-Or), enabled by MLA type m_2, and iii) all existing NFV-LOs between themselves (Lo-Lo), enabled by MLA type m_3.

The value calculated in (4.1) is smaller or equal than the maximum number of interfaces that can be established (e.g., all NFV-LOs from all edge and administrative domains are connected directly to each other, being at the same time connected to their respective NFV-SOs). In particular, the MLAs that enable the establishment of particular reference points in the orchestration platform can be considered as a triplet, i.e., (m_1, m_2, m_3). Such a triplet refers to a permutation of the three types of MLAs, i.e., m_1, m_2, m_3, which enable establishment of Or-Lo, Or-Or, and Lo-Lo, reference points, respectively. In Fig. 4.4, we illustrate the three examples of arrangement of the architectural elements (i.e., NFV-LOs and NFV-SOs), and pair them with the corresponding triplet. Each triplet in practice

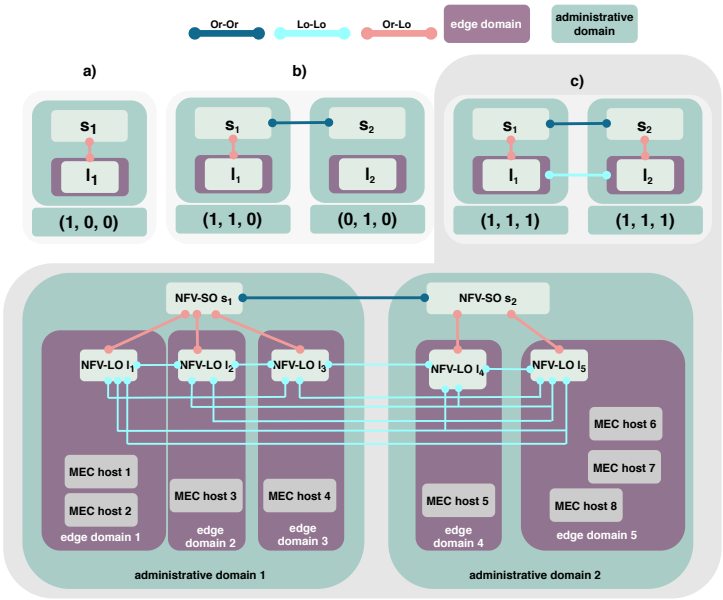


Figure 4.4: The example of hierarchical NFV management and orchestration in the orchestration platform for CCAM (Scenario III from Table 4.3)

means that certain permutation of agreements (i.e., MLAs) has been achieved between the top-level and edge-level orchestrators from different edge and administrative domains, thereby allowing edge-level orchestrators to consume resources from different domains to perform orchestration operations. In particular, the simplest scenario is shown in Fig. 4.4 a), which depicts the case when there is only one administrative domain, e.g., no collaboration between MNOs from different countries is present, and edge-level orchestrators are allowed to orchestrate only those resources that belong to their edge domains.

Both b) and c) in Fig. 4.4 depict the collaboration between the top-level orchestrators, but these two scenarios differ in terms of agreements between the edge-level and the top-level orchestrators, and between the edge-level orchestrators themselves. However, some of the triplets/permutations are not possible, such as $(m_1, m_2, m_3) = (1, 0, 1)$, because it is required to first establish federation between the top-level orchestrators, i.e., Or-Or reference points, in order to enable the direct Lo-Lo links between the edge-level orchestrators. This means that two NFV-LOs cannot cooperate via Lo-Lo link unless the federation between different administrative domains has been established. Hence, the complete list of MLA triplets for our orchestration platform is given as follows $(m_1, m_2, m_3) = \{(0,0,0), (0,1,0), (1,0,0), (1,1,0), (1,1,1)\}$, where the most complete case $((1,1,1))$, means that all instances of reference points are established between components in the architecture of our orchestration platform, as illustrated in example shown in Fig. 4.4 c).

Objective 1: To maximize utility function $U_1(i)$ that determines the number of available

instances of reference points in the orchestration platform.

$$\begin{gathered} U_1(i) = \sum_{s_1=1}^{N_S} \sum_{s_2=1, s_1 \neq s_2}^{N_S} x_{s_1 s_2} \cdot m_{2 s_1 s_2} + \\ \sum_{s=1}^{N_S} \sum_{l=1}^{N_L(s)} x_{sl} \cdot m_{1sl} + \sum_{l_1=1}^{N_L} \sum_{l_2=1, l_1 \neq l_2}^{N_L} x_{l_1 l_2} \cdot m_{3 l_1 l_2}, \\ U_1(i) \leq \frac{1}{2} \cdot (N_S(N_S-1) + N_L(N_L-1) + 2N_L), \\ x_{s_1 s_2} = \frac{1}{2}, \quad x_{l_1 l_2} = \frac{1}{2}, \\ d(l) = d(s) \rightarrow x_{sl} = 1, \\ d(l) \neq d(s) \rightarrow x_{sl} = 0. \end{gathered} \tag{4.1}$$

With reference to MLAs that are previously described in the form of triplets, we can draw an important conclusion about the number of hops that a request for a certain orchestration operation needs to pass to reach the final destination. For example, if NFV-LO from one administrative domain needs to extend the scope of application implementation that is running under its scope (i.e., the edge domain), it will send a request for application instantiation in other edge domain, either in the same or in other administrative domain. Thus, the level of agreement between the administrative domains, as well as the edge domains within their scope, defines the number of hops, i.e., n_h (equation (4.2)), which needs to be minimized in order to ensure lower latency while maintaining the service continuity.

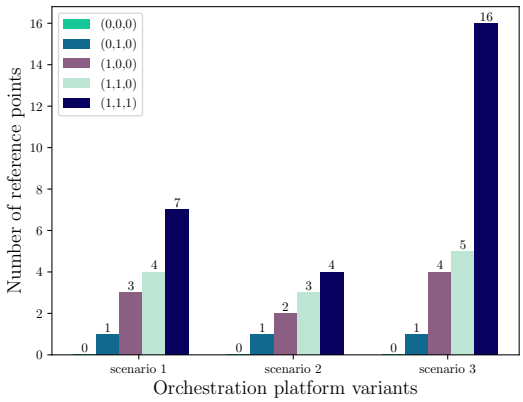
$$n_h = \begin{cases} 2, & m_1 = 1 \wedge m_3 \neq 1 \wedge d(l_1) = d(l_2) \\ 1, & m_1 = 1 \wedge m_3 = 1 \wedge d(l_1) = d(l_2) \\ 3, & m_1 = 1 \wedge m_2 = 1 \wedge m_3 \neq 1 \wedge d(l_1) \neq d(l_2) \\ 1, & m_1 = 1 \wedge m_2 = 1 \wedge m_3 = 1 \wedge d(l_1) \neq d(l_2) \end{cases} \tag{4.2}$$

Whether both edge-level orchestrators belong to the same administrative domain $d(l_1) = d(l_2)$, or to different administrative domains $d(l_1) \neq d(l_2)$, Fig. 4.5b shows the number of hops for an orchestration request from an arbitrarily defined edge-level orchestrator NFV-LO, which needs to reach another NFV-LO.

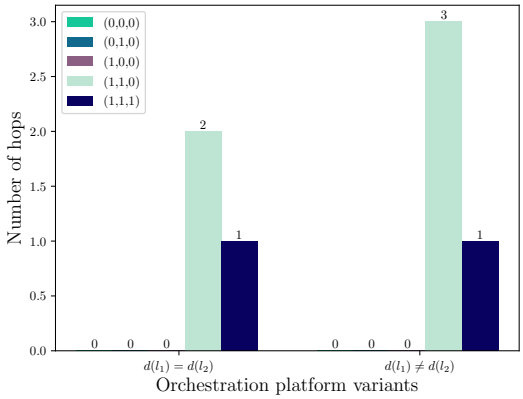
If we consider now the resources that can be assigned to a particular NFV-LO to perform orchestration operations, the variable x_{slin} represents a decision variable that is equal to one, if an instance of application implementation i has been assigned to l-th NFV-LO. Thus, the l-th NFV-LO can consume resources from n-th MEC host in s-th NFV-SO domain (i.e., MEC hosts that are available in NFV-SO domain). Otherwise, if the aforementioned combination is not allowed by MLA, the value of decision variable x_{slin} is equal to zero.

The amount of resources of type k that are available on n-th MEC host that is orchestrated by l-th NFV-LO, in s-th NFV-SO domain, is defined as ρ_{slnk}. Hence, if the federation and MLAs are agreed (either only Or-Or, or both Or-Or, and Lo-Lo, are established), the scope of resources, which l-th NFV-LO orchestrator is allowed to consume in order to perform orchestration operations, is extended to multiple domains.

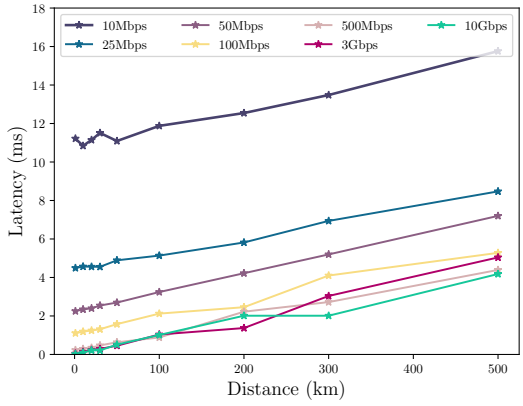
In inequation (4.3), the overall amount of resources of type k, which are given at disposal for performing orchestration operations on the i-th application implementation that is deployed



(a) Number of instances of reference points (scenarios I, II, and III).



(b) Number of hops for orchestration requests in case $d(l_1) = d(l_2)$, and $d(l_1) \neq d(l_2)$.



(c) Latency (transmission and propagation) depending on the network bandwidth.

Figure 4.5: Number of instances of reference points and number of hops for orchestration requests depending on different combinations of (m_1, m_2, m_3), and latency of transmission and propagation.

on the n-th MEC host, cannot exceed the maximum amount of available k-type resources on this node. Let us assume that the system consists of two NFV-SOs, as described in Scenario III from the Table 4.3, and illustrated in Fig. 4.4 c), thereby spanning three, and two edge domains, respectively. Each of these edge domains is orchestrated by one of the N_L NFV-LO orchestrators. If n-th MEC host is located in the domain of a particular top level orchestrator, i.e., the s-th NFV-SO, then the non-negative integer m_{1sl} determines whether l-th NFV-LO is allowed to consume k-type resources of n-th node located in the domain of s-th NFV-SO. Therefore, in the system that we previously described and illustrated in Fig. 4.4 c), if n-th MEC host is located in the domain of s_1 (e.g., MEC host 1, i.e., $n=1$), then $m_{1s_1l_1}$ determines whether NFV-LO l_1 can consume k-type resources from this node or not. For example, the sum member for the combination (s_2, l_1) is equal to zero in that case, because the resource will be already given to l_1 by s_1, as the selected MEC host is in the domain of s_1. Furthermore, for the combination (s_1, l_4), the sum will be non-zero in case there is at least Or-Or interface established between s_1 and s_2. Thus, all NFV-SOs allow any NFV-LO to consume resources from their domains, but if $m_1 = 1$ for the l_i-th NFV-LO that is not in the domain of s_j-th NFV-SO (i.e., $d(l_i) \neq d(s_j)$), this means that $m_{2s_js_{j*}} = 1$, i.e., the federation between the top-level orchestrators is established.

Objective 2: To maximize utility function $U_2(i)$ that determines the amount of resources, which are distributed in the hierarchical orchestration platform for CCAM, and given at disposal for performing orchestration operations on the application implementation i.

$$\begin{gathered} U_2(i) = \sum_{s=1}^{N_S}\sum_{l=1}^{N_L}\sum_{k=1}^{N_K} r_{slink}(m_{1sl}) \\ \sum_{s=1}^{N_S}\sum_{l=1}^{N_L} r_{slink}(m_{1sl}) \leq \rho_{nk} \end{gathered} \tag{4.3}$$

$$\sum_{s=1}^{N_S}\sum_{i=1}^{N_I}\sum_{n=1}^{N_H} x_{slin}(m_{1sl}) \cdot c_{ik} \leq \sum_{n=1}^{N_H} \rho_{nk}, \forall i \in I, k \in \mathcal{R} \tag{4.4}$$

The left side of inequation (4.4) expresses the amount of resources of type k, which are available in all MEC hosts (from all NFV-SO domains) and given at disposal to the l-th NFV-LO to perform orchestration operations. As it can be seen from (4.3) and (4.4), the resource availability is bounded by agreed level of MLA (i.e., m_{1sl}). The utility function illustrated by equation (4.5) models the overall utility $U_{slin}(x_{sl})$ that the system gains by assigning c_i resources to l-th NFV-LO, allowing it to deploy i-th instance of application implementation to its assignment vector x_{sl}. The assignment vector $x_{sl} \in \{0,1\}^{N_H \times N_I}$ refers to the combination of l-th NFV-LO and s-th NFV-SO, which has a task to deploy the instances of application implementations on top of the MEC hosts, and to perform orchestration operations on these instances.

Objective 3: To maximize utility function, which depends on: i) the MLAs that allow NFV-LOs to operate in a resource extended manner, which means that NFV-LOs can rely on the resources from other edge networks/domains to perform their orchestration operations, ii) the selected NFV-LO in particular NFV-SO domain to perform the orchestration operations, iii) the chosen implementation for the application instance, and iv) the selected MEC host for the deployment.

$$U = \sum_{s=1}^{N_S}\sum_{l=1}^{N_L}\sum_{i=1}^{N_I}\sum_{n=1}^{N_H} U_{slin}(x_{sl}) \cdot x_{slin}(m_{1sl}) \tag{4.5}$$

Therefore, aiming to achieve the *Objective 3* that refers to the overall orchestration platform for CCAM, there is a need to achieve *Objective 1*, and *Objective 2*, for all NFV-LOs in the orchestration platform.

4.1.7 Latency performance

Here we tackle the system model for describing latency performance over the Lo-Lo, i.e., the direct link between edge-level orchestrators (i.e., NFV-LOs). This direct link is used to transfer a request for any orchestration operation that is allowed to be requested or recommended from one local orchestrator to another, as described in Section 4.1. Henceforth, the overall transport network latency for such request can be defined as a cost function (equation (4.6)) that consists of the transmission delay, the propagation delay, the computational delay, and the queuing delay [145, 146]. If we define the request for orchestration operation as a_{orch}, where $a_{orch} \in \mathcal{O}$, $\mathcal{O} = \{1, \ldots, N_{orch}\}$, then the traffic that is generated by this request can be denoted as $f(a_{orch})$. Therefore, the transmission delay is defined as a time needed for processing an orchestration request on the transmitter side (NFV-LO in domain 1), and it can be expressed as a fraction of the traffic that this request generates and the bandwidth (B_{l_1,l_2}) of the processing link between two local orchestrators (i.e., l_1-th and l_2-th). As the fraction of the traffic and the capacity determines only the number of time-slots that are required by processing link to start transferring the request, it needs to be multiplied by a unit duration of a time-slot, i.e., t, in order to calculate the overall transmission delay. The propagation delay depends on the length of the link between two orchestrators, and the overall propagation speed over their communication link. The speed is determined as a speed of electromagnetic signal that is being transmitted over a certain medium, and it is usually calculated as speed of light, with the upper limit of 300000 km/s, which is the speed of light in a vacuum. In the overall transport network latency α_{l_1,l_2}, parameters β and γ are weighting factors that balance the networking characteristics [146], determining the variability of available bandwidth B_{l_1,l_2}, 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 orchestration hosts propagate over the transport network that is fiber-based, propagation is 1.5 times slower than in a vacuum, resulting in $\gamma = 1.5$.

$$\begin{aligned} \alpha_{l_1,l_2} &= \alpha_{t_{l_1,l_2}} + \alpha_{p_{l_1,l_2}} + \alpha_{c_{l_1,l_2}} + \alpha_{q_{l_1,l_2}} \\ \alpha_{l_1,l_2} &= \sum_{i,j \in \mathcal{L}_{l_1,l_2}} \beta \cdot \frac{f(a_{orch})}{B_{i,j}^{(l_1,l_2)}} \cdot t + \\ &\quad \sum_{i,j \in \mathcal{L}_{l_1,l_2}} \gamma \cdot \frac{l_{i,j}^{(l_1,l_2)}(m_{sl_1l_2})}{s} + \alpha_{c_{l_1,l_2}} + \alpha_{q_{l_1,l_2}} \end{aligned} \tag{4.6}$$

Let us consider that the link, which is used for transmission of the request for orchestration operation, is consisted of multiple segments, i.e., (i,j) with the length $l_{i,j}$, where $i,j \in \mathcal{L}_{l_1,l_2}$. In particular, $\mathcal{L}_{l_1,l_2}$ is the set of all link segments that can be chained to form the link between the local orchestrators l_1 and l_2, i.e., (l_1, l_2). The length of the link between two local orchestrator depends on the $m_{sl_1l_2}$ parameter, which determines whether there is a direct link between orchestrators or this link consists of the link segments that also include top-level

Table 4.4: System characteristics

System information		
Type	**Virtual Wall node**	**CityLab node**
Reference	Node 1	Node 2
CPU (GHz)	2.252	1
System memory	48 GB RAM	4 GB DDR3-1333 MHz
Processor	2x 8core Intel E5-2650v2 (2.6 GHz) CPU	AMD GX-412TC 1GHz Quad-core CPU
Storage	250 GB	240 GB
Disk	250GB HGST HTS725025A7	240GB Samsung SSD 850

orchestrators in the chain. Thus, taking into account the definition of the parameter $m_{sl_1l_2}$ ($m_{1l_il_j}$, $m_{2l_il_j}$, or $m_{3l_il_j}$), it is intuitive to conclude that the overall length of the link between NFV-LOs, i.e., l_{l_1,l_2} will be larger if the request from one local orchestrator needs to be passed to the top-level orchestrators first, and not directly via the Lo-Lo link. With regards to the aforementioned, the main contributing factors to the overall transport network latency are the network bandwidth (B_{l_1,l_2}), and the distance that orchestration operation request needs to propagate to reach NFV-LO 2 from NFV-LO 1, or vice versa. In Fig. 4.5c, we show the latency that consists of transmission and propagation delay defined in equation (4.6), which is calculated for a simple orchestration request (simple request carrying 13KB of data, as described in Section 4.1.8), depending on the bandwidth of the network links, and the distance between respective orchestrators. Hence, the latency will be higher in case of the lower network bandwidth, but also in the case of larger distances between local orchestrators (i.e., links NFV-LO 1 - NFV-SO 1 - NFV-SO 2 - NFV-LO 2, or NFV-LO 1 - NFV-LO 2). For example, Maheshwari et al. [145] show that the average response time of servers in cloud and edge also increases with an increase in system load, which is of course affected by computation, i.e., processing of the request on the orchestrator side in our case. Thus, in Section 4.1.8, we assess the overall latency, i.e., the average response time for orchestration requests generated by edge-level orchestrators.

4.1.8 Experimental assessment of the orchestration platform

In this section, we present the experimental assessment of the orchestration platform for collaborative edges, thereby demonstrating the relevance of the design choices that we made for the architecture itself, but also for the operational aspects of such orchestration platform. In order to conduct experiments that assess their performance, as well as the capacity to perform the orchestration operations, we defined and performed a set of tests for both top-level and edge-level orchestrators in the testbed environment.

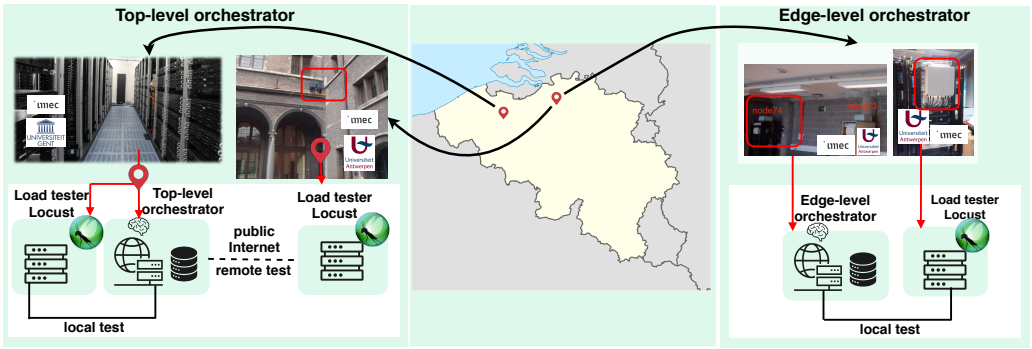


Figure 4.6: The nodes used for local and remote tests on top of the Virtual Wall and CityLab testbeds.

4.1.8.1 Testbed environment

The system characteristics of computing machines that we used in experimentation are listed in Table 4.4. Taking into account the characteristics of the top-level orchestrator (presented in Section 4.1), it is expected to run on top of the resourceful computing machines (such as Node 1 in Table 4.4), as it serves all underlying edge domains while covering the whole administrative domain (e.g., one country). Thus, in our experiments, we leveraged on the computing capabilities of the Virtual Wall[4] testbed [147] (Fig. 4.6) for the purpose of testing the response to orchestration requests of the top-level orchestrator, as well as to evaluate its average load. The Virtual Wall testbed, located in Gent, Belgium, consists of more than 550 powerful bare metal and GPU servers, which are software and hardware configurable, i.e., configurable in terms of software installation (e.g., operating systems, and drivers), and networking via configuring the physical interconnection between network interfaces. All these machines forge a generic environment for advanced networking, distributed software, cloud, big data, and scalability research and testing.

On the other hand, the edge-level orchestrator is designed to cover smaller areas, i.e., edge domains, while performing management and orchestration operations of the deployed edge services, but also responding to the requests that are coming from adjacent or other edge domains. Thus, for testing the capabilities of an edge-level orchestrator, we utilized the CityLab[5] testbed [148] (Fig. 4.6), i.e., the resource constrained Node 2 presented in Table 4.4. In particular, CityLab is a smart city large-scale wireless networking testbed, which is located in Antwerp, Belgium, whereas the experimentation nodes are attached to buildings and streetlamps providing the opportunities for experimentation at a city neighborhood level in the unlicensed spectrum. We have used public internet to establish the connectivity between different orchestration entities in this testbed environment, i.e., Lo-Lo reference point between edge-level orchestrators, and Or-Lo between top-level and edge-level orchestrators.

[4]Virtual Wall testbed: `https://www.ugent.be/ea/idlab/en/research/research-infrastructure/virtual-wall.htm`

[5]CityLab testbed: `https://doc.lab.cityofthings.eu/wiki/Main_Page`

Table 4.5: Description of the tests.

Type of machine	**Platform component**	**Test**	**Type of request**	**Average content size (B)**	
Node 1	Top-level orchestrator	Local	simple - only GET	13	
			only GET	400	
			GET & PUT	GET	140
				PUT	54
		Remote	simple - only GET	13	
Node 2	Local orchestrator	Local	simple - only GET	13	
			only GET	400	
			GET & PUT	GET	140
				PUT	54

4.1.8.2 Types of tests

For both the top-level and edge-level orchestrators, we performed different types of tests, as described in Table 4.5. The local tests refer to the tests in which server (i.e., the orchestrator) and client (i.e., load testing tool) are deployed on top of the two separate bare-metal machines that are connected by wire. Accordingly, in the remote test, server and client are dislocated, and there is an additional contributor to the overall latency, which is imposed by sending orchestration requests via public Internet (Fig. 4.6).

For the local tests, we conducted experiments with different test variants, which differ in complexity of the orchestration request. With reference to the software design of our orchestration platform presented in Section 4.1, each orchestration request is generated, received, and processed, as a REST API request. Therefore, we differentiate the complexity of different requests by performing: i) only simple GET requests, containing relatively small body (i.e., average content size), ii) only GET requests that involve certain transactions and checkups in database, and iii) a combination of GET and PUT requests, where PUT requests usually refer to those requests that require changes in the service deployments, reflected by applying changes in database as well. We designed the combined test in a way it generates three times more GET requests than PUT requests, as there are usually more query types of orchestration requests, where different orchestration entities ask other orchestrators about the state of a deployed service, and some of its particular parameters, than those requests that involve actions on application/service as an outcome of the orchestration algorithms (e.g., scaling up/down/in/out).

4.1.8.3 Load tester Locust

To generate orchestration requests and test performance of the orchestrators, we used a python-based performance testing tool Locust[6]. As Locust is widely used for performing stress and load tests on web servers, it is suitable for our experiments as both top-level and

[6]Locust: `https://docs.locust.io/en/stable/`

edge-level orchestrators are here deployed and tested as web-based servers, using python web framework Flask.

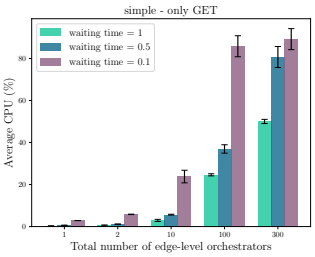
Locust enables defining the behavior of users in a regular Python code, running each of the users inside its own greenlet, i.e., a lightweight process, without the need for using callbacks. In the case of top-level orchestrator, users that generate requests are its underlying local, i.e., edge-level orchestrators, which are sending the orchestration requests. Similarly, in the case of edge-level orchestrators, users are other (adjacent or not) edge-level orchestrators that are directly connected to each other via low-latency Lo-Lo link.

4.1.8.4 Metrics

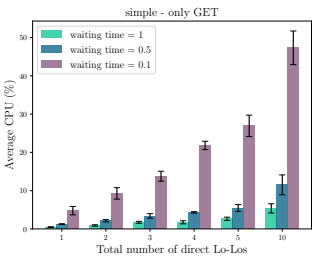
In all the tests that are executed, a several important KPIs are measured, which are relevant because they reflect the capability of an orchestrator to perform orchestration operations efficiently, as well as the amount of resources that it consumes for its work. These KPIs are: i) average response time per orchestration request, ii) average Central Processing Unit (CPU) load, iii) average Random-Access Memory (RAM) load, and iv) average power consumption, and they are described as follows. For instance, the average response time of both top-level and edge-level orchestrators is the overall latency of performing a particular orchestration request, from the moment when the request is generated in the edge-level or top-level orchestrator, to the moment when this request is processed. With reference to our analytical model presented in Section 4.1.5, the latency performance model includes an orchestration request a_{orch} that generates a certain amount of traffic $f(a_{orch})$. In the case of this evaluation, the response time in remote tests also includes the propagation and transmission latency as a result of sending an orchestration request a_{orch} through the communication link between two orchestrators, as well as the aforementioned time to process the upcoming traffic $f(a_{orch})$.

We need to make sure that both the top-level and edge-level orchestrators can handle the load of orchestration requests. Hence, the CPU and RAM load refer to the load that orchestrator can expect and experience when certain number of orchestration requests are received, which is a direct implication of the MLAs that determine a number of established interfaces between orchestration entities as described in Section 4.1.6 (Objective 1). The goal of measuring these KPIs is to assess the average behavior of both resourceful, and resource-constrained machines, which can host top-level and edge-level orchestrators, respectively. As we presented in Section 4.1.5, in case no direct link between edge-level orchestrators is established by MLA, it ultimately results in an increase in number of orchestration requests towards the top-level orchestrator. That is why in our tests we aim to assess the impact of such increase in number of requests, on the performance of the top-level orchestrator, and to evaluate the burden it imposes to the operations in the lop-level orchestrator.

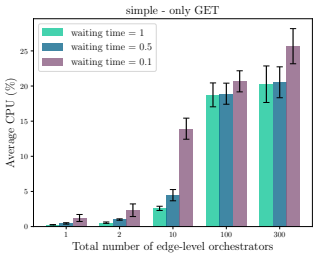
In the experimental evaluation we also measure the average power consumption of the top-level and the edge-level orchestrators, while they are performing orchestration requests. Since energy efficiency is considered as one of the ultimate goals of 5G ecosystem [149], the applications and processes that are executed on the edge and cloud computing devices need to be energy efficient. According to the European Commission's final study report on energy efficient cloud computing technologies [150], the design of any application has a high



(a) Top-level orchestrator, local test.

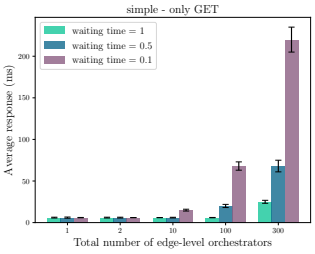


(b) Edge-level orchestrator, local test.

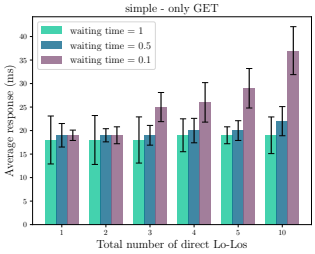


(c) Top-level orchestrator, remote test.

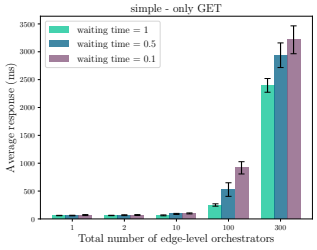
Figure 4.7: Average CPU load in Simple - only GET test.



(a) Top-level orchestrator, local test.



(b) Edge-level orchestrator, local test.



(c) Top-level orchestrator, remote test.

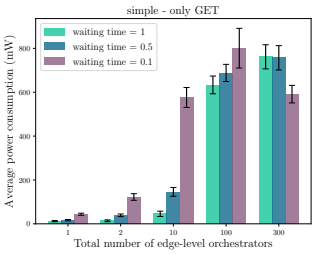
Figure 4.8: Average response time per orchestration request in Simple - only GET test.

impact on its energy consumption. This becomes even more evident when similar applications may require different consumption of CPU load, and memory load, and ultimately different energy consumption. Therefore, in [150] it is stated that software is a major factor for energy-efficiency when the energy consumption is measured for a cloud computing product. Thus, it is important to measure the impact of orchestration operations on the energy and power consumption, thereby designing orchestration solutions to be low energy consuming techniques.

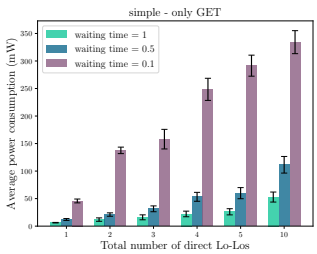
The experiments that we described in this section enabled us to evaluate a relative average response time, and CPU/RAM load, and average power consumption, for orchestration requests that originate at the edge-level orchestrator for example, and terminate on another edge-level orchestrator in case there is a direct link between these two edge-level orchestrators, and in case the orchestration requests need to be forwarded via top-level orchestrators. All the results that we present in the following section reflect the relative behavior of orchestration entities within our orchestration platform, because this behavior depends on the type of machine that hosts the orchestrator, the type of the orchestrator, and the complexity of orchestration operations that this orchestrator performs.

4.1.8.5 Results

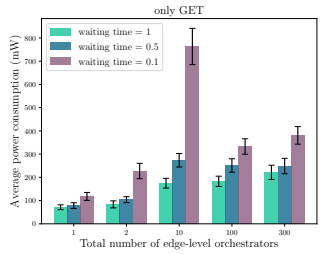
Let us consider a scenario with one top-level orchestrator per whole administrative domain (e.g., country), and multiple edge-level orchestrators, with no direct Lo-Lo link established



(a) Top-level orchestrator, local test.



(b) Edge-level orchestrator, local test.



(c) Top-level orchestrator, local test.

Figure 4.9: Average power consumption in Simple - only GET test (a and b), and only GET test (c).

Table 4.6: Average RAM load.

Orchestrator	**Type of test**		**Waiting time (s)**		
			1	0.5	0.1
			Average RAM (%)		
Top-level orchestrator	local	simple - only GET	0.2	0.2	0.2
	local	only GET	0.5	0.5	0.5
	local	GET & PUT	0.2	0.51	0.68
	remote	simple - only GET	0.2	0.2	0.2
Edge-level orchestrator	local	simple - only GET	0.7	0.7	0.7
	local	only GET	0.8	0.8	0.8
	local	GET & PUT	0.8	0.8	0.8

between them as per definition in equation (4.1) (Section 4.1.6). In such case, all the traffic that edge-level orchestrators generate in their domains by sending orchestration operation requests towards other edge domains, first reaches their responsible top-level orchestrator. In Figures 4.7a and 4.8a, and Table 4.7, we can clearly see the increasing trend in CPU load and average response time, respectively, for the top-level orchestrator with the number of edge-level orchestrators that are simultaneously sending orchestration requests towards it. The same trend applies to the edge-level orchestrators (Figures 4.7b and 4.8b, and Table 4.8) with the increase in total number of direct Lo-Lo connections.

For each total number of edge-level orchestrators, and direct Lo-Lo connections, shown on the x-axis of all graphs shown in Figures 4.7, and 4.8, and Tables 4.7, and 4.8, we run tests for different waiting time between successive requests that are coming from a single edge-level orchestrator. It means that in case of waiting time equal to 1s, each edge-level orchestrator is generating one orchestration request per second. Accordingly, each of them is generating 10 orchestration requests per second in case of waiting time equal to 0.1s. Therefore, in case there are 100 edge-level orchestrators distributed across a single administrative domain, the top-level orchestrator needs 68 ms in average to process a simple orchestration request (e.g., response to a query about resource availability in a certain edge domain). In practice, that means that edge-level orchestrator will wait 68 ms only for the first top-level orchestrator to process its request, which will then include also an additional latency that propagation and transmission of this request, as per equation (4.6) in Section

4.1.5, take from i) local to the top-level orchestrator, ii) from the top-level orchestrator in domain 1 to the top-level orchestrator in domain 2, and iii) from the top-level orchestrator in domain 2 to the target edge-level orchestrator in domain 2. On the other hand, if a direct Lo-Lo link is established from originating to the target edge-level orchestrator, Fig. 4.8b shows that one edge-level orchestrator (although resource constrained) will take only 19 ms to process the same orchestration request. Taking into account the propagation latency in equation (4.6), we can assume that the overall latency via link Lo-Lo will be lower than in case when request is sent through the top-level orchestrators (Section 4.1.7), which in total results in a at least three times lower latency in processing orchestration request in case of having Lo-Lo link.

If we now reflect on the remote test for the top-level orchestrator, which is depicted in Fig. 4.6, the increase in average response time per orchestration request can be seen in Fig. 4.8c in comparison to Fig. 4.8a. For example, in the case 10 edge-level orchestrators are simultaneously sending two requests per second towards the top-level orchestrator, we can see that in remote test the average response time is 527 ms while being only 20 ms in the local test. Such an increase in average response time is expected due to delay in sending orchestration requests via public internet, as well as queuing in the gateways, highly depending on the number of the network links between orchestrators, their length and of course bandwidth. As such result might severely disrupt the performance of vehicular applications, especially the latency constrained ones, due to the increase in orchestration execution, we emphasize the importance of the direct low-latency Lo-Lo links that should significantly decrease the overall delay. A further reduction of the latency, caused by congested network nodes, can be achieved also by dedicating more processing power, or more network adapters, to a particular orchestrator.

Tackling the load that the top-level orchestrators need to handle in case the MLA do not allow edge-level orchestrators to directly collaborate via low-latency links, we assess the average CPU load (Fig. 4.7, and Tables 4.7, and 4.8), as well as the average RAM load (Table 4.6). The average RAM load remains stable in all tests, being slightly increased with complexity of orchestration requests, whereas the average CPU load is highly affected by the amount of orchestration tasks to process. In particular, Figures 4.7a and 4.7b, and Tables 4.7 and 4.8, show that for both top-level and edge-level orchestrators, the average CPU load increases with the number of edge-level orchestrators generating requests, and with the number of requests per second. One specific case when this load decreases is the GET & PUT test, in which the average CPU load for 100 and 300 edge-level orchestrators (Table 4.7) is smaller than in case of less complex tests (Fig. 4.7a). This decrease happens due to request queuing that significantly increases average response time (Table 4.7), which also results in failed requests, i.e., with rate of 2.27% for GET, and 7.35% for PUT requests, in case of 100 edge-level orchestrators, and 8.27% for GET, and 35.52% for PUT requests, in case of 300 edge-level orchestrators. To measure the average power consumption of different orchestration components while performing orchestration operations, the same set of experiments has been executed for local tests as for measuring the average response time and CPU/memory load. We have utilized the Linux-based command-line program PowerTOP[7], which provides an estimate of the total power consumption of the overall system, but also individual power consumption for individual processes, devices, kernel workers, etc. The obtained results are shown in Fig. 4.9, and we can see an increasing trend in average power

[7]Managing Power Consumption with PowerTOP: `https://red.ht/2T9ZF3z`

Table 4.7: Results for the top-level orchestrator in local tests.

Top-level orchestrator	**only GET test waiting time**			**GET & PUT test waiting time**		
	Average CPU load (%)					
Total number of local orchestrators	1	0.5	0.1	1	0.5	0.1
1	0.4	1	3.6	0.4	0.8	2.9
2	0.8	1	7.9	0.8	1.3	6.4
10	3.8	8.6	29.7	3.9	23.9	24
100	31.9	52.9	123.8	19.9	24.2	26.1
300	71.7	114.4	127.9	34.1	35.8	39.7
Average response time (ms)						
1	7	7	7	17	18	18
2	7	10	10	20	20	22
10	7	10	10	23	27	70
100	10	17	68	370	64	1139
300	35	69	408	2050	2238	2410

consumption of both top-level and edge-level orchestrators with the increasing number of orchestration requests in the simple - only GET test. However, when number of edge-level orchestrators that simultaneously send orchestration requests towards the top-level orchestrator increases above 100, we can see that average power consumption drops. The same happens also in the case of more complex orchestration operations, as Fig. 4.9c shows. As described for CPU and memory load, average power consumption also decreases due to the request queuing that significantly increases average response time.

4.1.8.6 Discussion

With reference to analytical evaluation of the collaborative orchestration platform in Section 4.1.5, and its experimental assessment in Section 4.1.8, here we briefly pinpoint a few main aspects to consider for an orchestration platform that reinforces the orchestrated mobile edge networks.

Number of instances of reference points impacts the communication delay and resource availability The number of available instances of reference points (equation (4.1), Fig. 4.5a) in the orchestration platform reduces the overall number of hops (equation (4.2), Fig. 4.5b) that certain orchestration request, originating from an edge-level orchestrator, needs to pass in order to reach target edge-level orchestrator. Thus, such number of instances of reference points needs to be increased, as it not only reduces the communication delay in orchestration requests but it also increases the amount of available resources, given to each edge-level orchestrator at disposal to efficiently perform orchestration operations (equation (4.3), with the maximum amount of available resource in all domains expressed by inequation (4)). Considering diversity in resource availability on the edges from the same or different administrative domains, it is important for orchestrators to have more resources

Table 4.8: Results for the local orchestrator in local tests.

Local orchestrator	**only GET test waiting time**			**GET & PUT test waiting time**		
	Average CPU load (%)					
Total number of direct Lo-Los	1	0.5	0.1	1	0.5	0.1
1	0.9	1.7	7.7	1.02	1.96	8.83
2	1.7	3.6	17.3	2.67	3.9	18.23
3	3.4	5.4	26.3	2.99	7.21	23.3
4	3.6	8.5	33.9	4.04	9.01	34.74
5	4.3	10.2	42	5.68	11.35	42.71
10	10.4	21	89.7	11.29	22.05	79.9
Average response time (ms)						
1	22	22	23	25	25	25
2	22	23	26	25	25	30
3	22	23	27	26	27	30
4	22	24	32	26	27	35
5	23	24	32	26	32	43
10	23	30	56	27	32	61

at disposal for deploying CCAM services. On the contrary, if orchestrators running on different edges do not establish agreements for collaboration, CCAM services might suffer from performance degradation due to the limited amount of resources at the available edges. Due to the lack of resources in its own domain, an orchestrator might not be able to deploy e.g., a relevant safety CCAM service (e.g., change the lane warning, brake warning, slow down warning) that needs to support emergency situations on the road.

Number of instances of reference points impacts the orchestration load The negotiated MLAs increase the number of used instances of reference points, thereby significantly reducing the load of the top-level orchestrator, as the upcoming requests from the edge-level orchestrators do not need to be transferred via the top-level orchestrator to other edges. Otherwise, the increase in number of requests increases the CPU load (Fig. 4.7a), which then causes a significant increase in average response time per orchestration request (Fig. 4.8a). Such an increase in average response time might significantly delay e.g., the instantiation of a CCAM service, or any runtime operation such as scaling up/out. Let us consider that vehicle is driving on the highway with the speed of 80km/h, thereby consuming the CCAM service that sends notifications about the conditions on the road. If CCAM service is unavailable due to the scale-up operation, which is triggered to improve service resource utilization and decrease service latency, and if scaling-in lasts for approximately 500ms, it will result in at least 500 ms delay in road information update. Such an increased response time will imply an outdated or delayed notification sent to the vehicle that needs to change its manoeuvre, i.e., vehicle will already pass the additional 11,1m, which can prevent it from changing the lane in time.

Direct Lo-Lo links impact the average response time Given the aforementioned importance of the average response time of orchestration for the CCAM services, the design choices might include more direct links between edge level orchestrators to decrease the response time, i.e., to fasten the runtime orchestration operations such as scaling and service relocation (Objective 3, i.e., equation (5)). Although deployed on resource-constrained edge clouds, if low-latency links are established, and used as per MLA, the edge-level orchestrators process the orchestration requests with a reduced average response time comparing to the top level orchestrators (Fig. 4.8), due to i) the decreased load, and ii) the decreased propagation and transmission latency over the direct link (Fig. 4.5c). With respect to results presented for the average response time, and CPU load, one reasonable design choice for the orchestration can enforce using direct Lo-Lo links for those orchestration operations that directly affect the runtime of the service (e.g., scaling from the previous example, or service migration), while other operations such as instantiation/termination can be performed via top-level orchestrators to balance the load properly.

Orchestration operations impact the overall power consumption on the edges Albeit neither the top-level orchestrators, nor the edge-level orchestrators, are intended to run on low-energy IoT devices, their power usage is still relevant for the overall energy consumption plan in the 5G ecosystem, especially due to the evident increase in consumption with the increase and complexity of orchestration requests. As shown in Fig. 4.9a, average consumption increases for more than 100mW in case number of edge-level orchestrators increases from two to 10, with two requests per second from each. Thus, balancing the orchestration load across multiple edge-level orchestrators is essential, as it also balances the energy consumption across edges, making the resource and service orchestration an energy-aware technique for 5G ecosystem.

Orchestrators' response time affects the service continuity We learnt that it is important to carefully consider the number of hops presented in Section V, as it significantly impacts the average response time per orchestration request, which is also seen in the results presented in Section 4.1.8. The load on the orchestrators needs to be balanced in order to keep their response time low. As we presented in Section 4.1.3, and illustrated in Fig. 3, achieving edge-to-edge service continuity is possible if orchestration entities i) deploy a peering service instance in the target domain towards which the vehicle is driving, ii) relocate the application state from the source to the target domain, and iii) relocate the user endpoint to the target application instance. All these operations are performed by the orchestration entities, thus, their response time is critical for achieving timely relocation of the service, and maintaining service continuity when vehicle is moving from one domain to another.

4.2 Summary of the Chapter

In this Chapter, we proposed a comprehensive multi-tier orchestration framework for 5G-enhanced vehicular systems where application services are serving highly mobile users, and as such are deployed as edge services/EdgeApps at the network edge, i.e., closer to those

users with stringent service requirements. Taking into account the gaps identified in existing MANO systems (Chapter 3), we designed this orchestration framework to enable continuity of low-latency edge services and EdgeApps running at the distributed network edges, while associated users are traversing from one network domain to another. The key to achieve such an efficient service orchestration lays in maintaining the collaboration between edge orchestrators that are managing smaller pieces of the overall 5G ecosystem, i.e., their respective edge domains. We have defined the design principles of such orchestration systems, focusing on the proactive deployment of edge services/EdgeApps (i.e., multi-edge service deployment), and edge-to-edge service continuity. To evaluate the performance of such orchestration systems in vehicular environments, we defined the analytical model, as well as the real-life experimentation setup for collecting performance results. With the analytical and experimental evaluation, we draw conclusions on the gain in accelerating orchestration operations while balancing associated protocol and computational load over the distributed and multi-layered orchestration platforms.

Given the importance of orchestration delay that we introduced early in this thesis (Chapter 2), in the performance analysis within Chapter 4, we studied the impact of orchestration platform on the communication delay in average response time of an orchestration request. We showed that the number of reference points established within the orchestration elements (within the same or different orchestration layers) impacts the communication delay as well as availability of resources that can be used for service deployments on the network edges. In particular, if there is a direct edge-to-edge reference point established between adjacent edge orchestrators, the average orchestration delay (i.e., response to orchestration request) could be decreased three times compared to scenarios with no direct reference points between edge orchestrators. We have also studied the impact of orchestration load on the orchestration delay, and we learned that number of edge orchestrators that a single top-level orchestrator needs to handle, impacts its orchestration load. In that case, the CPU load increases by approximately 70% if a number of edge orchestrators increases from 1 to 10, resulting in average top-level orchestrator's response time of 500 ms. Such an increase in orchestration delay is usually not acceptable for vehicular edge services as spending 500 ms without service response could lead to significant service deterioration. Thus, enabling direct reference points between edge orchestrators is encouraged, as it reduces the overall orchestration delay due to the decreased CPU load of single orchestrators, and the reduced transmission and propagation latency. Finally, to make future orchestration systems able to handle distributed edge service and EdgeApp deployments in an energy efficient manner, it is important to pursue energy-aware orchestration mechanisms. Given the increase of approximately 90% in power consumption at the top-level orchestrator's level when the number of underlying edge orchestrators increases from 1 to 10, it is important to employ load-balancing mechanisms at different orchestration levels to avoid excessive increases in power and energy consumption.

Chapter 5

Orchestrated EdgeApps as a 5G booster for automotive, and transport & logistics services

This chapter is part of the **Contribution 3**: *Orchestrated Edge Network Applications (EdgeApps)*, and it is based on:

N. Slamnik-Krieštorac, G. Landi, J. Brenes, A. Vulpe, G. Suciu, V. Carlan, K. Trichias, I. Kotinas, E. Municio, A. Ropodi, and J. M. Marquez-Barja, "Network Applications (NetApps) as a 5G booster for Transport & Logistics (T&L) Services: The VITAL-5G approach," 2022 *IEEE European Conference on Networks and Communications (EuCNC)*, 2022.
N. Slamnik-Krieštorac, F. Z. Yousaf, G. M. Yilma, R. Halili, M. Liebsch, and J. Marquez-Barja, "Edge-aware Cloud-native Service for Enhancing Back Situation Awareness in 5G-based Vehicular Systems *(Submitted)*," in *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022.

In this Chapter, we present the concept of EdgeApps, as virtualized network functions that are designed i) to abstract the complexity of vertical services that stretch over automotive and T&L industries, and ii) to make vertical services able to leverage benefits of the underlying 5G network infrastructure. First, we provide some general insights into the concept of EdgeApps, showcasing examples of how the EdgeApps can be designed and chained into vertical services for enhancing T&L operations in river/sea ports in Section 5.2. Second, we study and present in detail a vertical service for enhancing mission-critical operations on the roads by creating an extended awareness of emergency vehicles, where the building blocks of such an 5G application service can be considered as EdgeApps as well.

The 5G ecosystems usually consists of 5G New Radio, 5G Transport network, 5G Core, and virtualized edge and cloud infrastructure. As such, they are enabling ultra-low latency (1-10 ms), ultra-high reliability (99.999%), and high data rates (up to 20 Gbps) [151], by creating logical and virtualized networks, i.e., network slices, over the common network infrastructure. Thus, by implementing uRLLC, eMBB, and mMTC, 5G expands the perspectives for industry

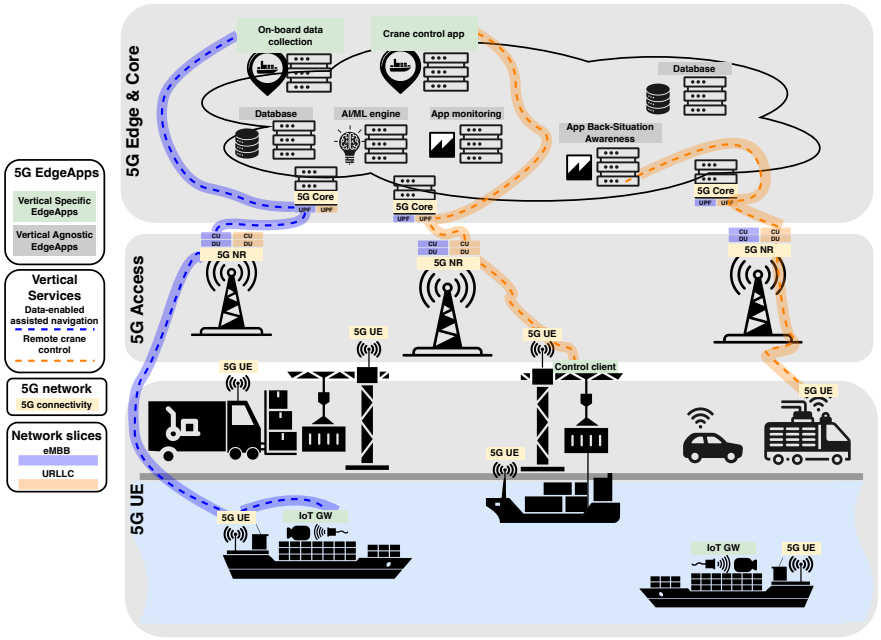


Figure 5.1: 5G EdgeApps as building blocks of T&L and automotive vertical services for providing faster and safer port/road operations in 5G ecosystem.

verticals such as automotive, e-health, and T&L systems, and it fosters new use cases (e.g., autonomous driving, remote navigation, teleoperation) that have not been possible with the previous generations of mobile communications systems, given the too stringent connectivity requirements for those use cases [152].

Thus, to be able to benefit from 5G, the design of vertical services needs to be tailored to particular use cases, taking into account vertical service-specific requirements towards 5G (e.g., service interruption for automated vessel control needs to be lower than 150ms [153]). To this end, in this paper we define the concept of EdgeApps, as a fundamental building block of the T&L service chains that are deployed on top of the 5G-enabled infrastructure (as illustrated in Fig. 5.1). The goals of breaking a complex vertical service to EdgeApps are: i) to simplify the composition of such vertical service chain, ii) to better describe the service-level information (vertical specific), iii) to specify 5G-related requirements for this service (e.g., 5G slices, 5G Core services), and iv) to abstract the underlying complexity, and thus to bridge the knowledge gap between vertical stakeholders, the network experts, and the application developers. The aforementioned is achieved by extending the orchestration-oriented models proposed by ETSI NFV, i.e., VNFDs and NSDs, which are service-agnostic, and limited to internal network service structure (i.e., the definition of computing resources, network functions in the chain, forwarding graphs and paths, virtual links, and internal/external connection ports). Such gaps in current standards can be bridged by adopting the EdgeApp modelling, i.e., through the declaration of i) protocols and languages used at the service interfaces of applications, ii) dependencies on hardware and devices, and iii) requirements on 5G mobile connectivity or 5G core network services.

In the following section, we define the concepts and modelling of 5G-enabled EdgeApps, and

categorize those EdgeApps depending on their specific features and vertical needs, as a work carried out in the scope of the VITAL-5G project [154]. Afterwards, in Sections 5.2 and 5.3, we showcase the applicability of EdgeApps for providing faster and safer operations of vessels (T&L), and for improving back-situation awareness on the highways (automotive), respectively.

5.1 The Concepts and Modelling of 5G-enabled EdgeApps

5.1.1 Packaging and management of EdgeApps

The EdgeApp concept facilitates the creation, design, provisioning, life-cycle management, and performance evaluation, of vertical services in 5G network infrastructures. A EdgeApp is a 5G-enabled virtual application which provides its own set of functionalities when deployed as a stand-alone entity, capable to cooperate and to interact with other EdgeApps to deliver more complex vertical services. In this sense, a EdgeApp can be considered as an atomic component of vertical services, which can be dynamically instantiated in multi-tenant virtual environments, re-used, composed, and shared, in the context of multiple service chains, as well as combined with 5G network slices to guarantee the required performance for the mobile connectivity (e.g., required uplink bandwidth for camera streams, and end-to-end latency for control signals towards vessels).

EdgeApps are derived from the concept of the ETSI VNFs, inheriting their capability to be automatically provisioned, scaled, terminated, monitored, and re-configured, in a multi-tenant virtual infrastructure through the creation and management of Virtual Machines or containers, as defined in their VNF packages [155]. In particular, EdgeApps extend the original VNF concept declaring i) service level information to simplify their distribution, sharing, and integration in vertical services, and ii) mobile connectivity requirements in terms of 5G network slice profiles or consumed 5G core services to automate their instantiation in 5G networks. This additional information is encoded as metadata in a EdgeApp blueprint, which is included in the EdgeApp package.

As illustrated in Fig. 5.2, the EdgeApp package includes i) the references to the VNF package that defines how to orchestrate the EdgeApp in an NFV MANO environment, ii) the EdgeApp blueprint, and iii) the additional elements like software licenses, software documentation, test cases, and target KPIs for automated validation. The EdgeApp packages can be on-boarded, searched, and visualized through an online repository, such as VITAL-5G Open Online Repository [156]. This repository provides an open catalogue of EdgeApps which can be provided by different developers and combined to deliver new services. This approach will facilitate the sharing of EdgeApps produced and distributed by different software developers.

Fig. 5.3 provides a graphical representation of the EdgeApp modelling, using as example a EdgeApp for management of IoT devices reachable via 5G network. In this example, the EdgeApp handles IoT data from/to IoT supervisors acting as IoT gateways installed in the field (e.g., in a vessel) and interconnected via 5G to the virtual computing infrastructure where the EdgeApp is running. A EdgeApp is composed by a set of internal Atomic Components (the red boxes), which correspond to containers or VMs implementing parts of the

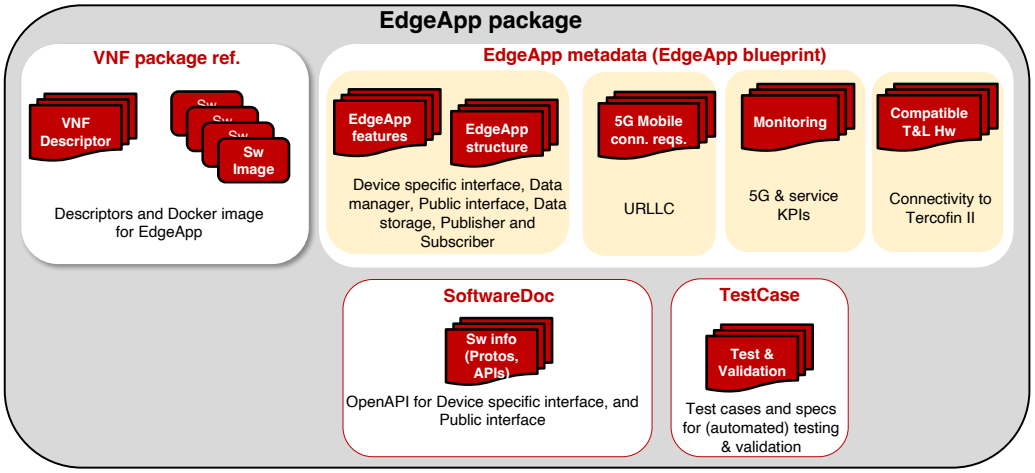


Figure 5.2: High-level EdgeApp package representation

EdgeApp logic. These components interact via internal Connectivity Services (the dotted line in the EdgeApp box), which correspond to virtual networks that connect their endpoints. The endpoints can be internal ones (light-grey circles), used only for intra-EdgeApp interactions, or external ones, used to interact with external entities (e.g., other EdgeApps, end users, or hardware elements such as devices installed in the vessels).

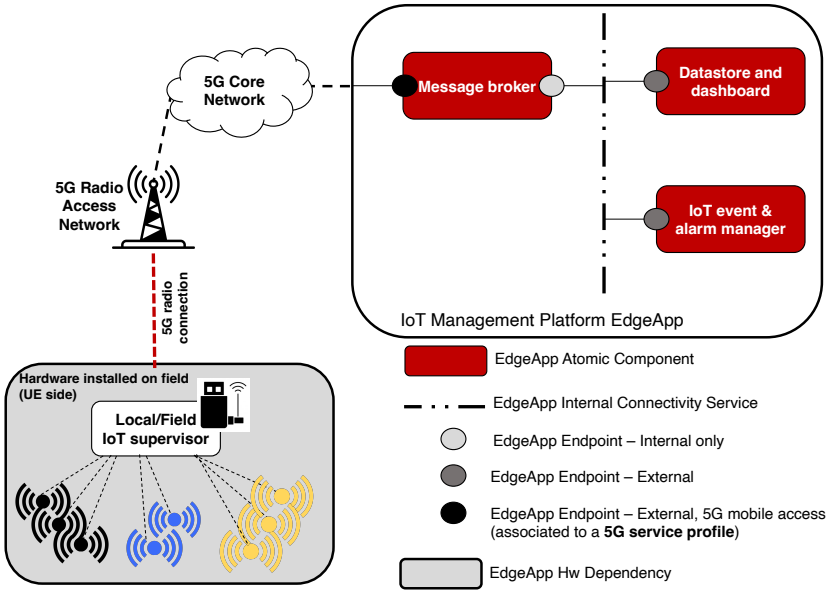


Figure 5.3: The example of EdgeApp representation.

The external endpoints that connect EdgeApps with the 5G network, using the N6 interface[1] of the 5G system [138], are characterized by additional attributes that describe the

[1]In the 3GPP 5G architecture, N6 reference point is connecting User Plane Function (UPF) in the 5G Core, with the Data Network (DN), where the EdgeApps are deployed in our case.

Table 5.1: EdgeApp classification.

	component and deployment methodology	
intended service	vertical-specific component-based	vertical-specific service-based
	vertical-agnostic component-based	vertical-agnostic service-based

mobile connectivity requirements for the EdgeApp traffic in uplink and downlink. These endpoints are thus associated to one or more 5G slice profiles, describing the network slice characteristics, as defined in the 3GPP Network Resource Model [157]. Some examples of the attributes are the slice service type (eMBB, uRLLC, or mMTC), QoS parameters (e.g., uplink and downlink data rate, latency, jitter), coverage area, and radio access technology. Moreover, the EdgeApp model describes the 5G network services consumed by the EdgeApp, e.g., the network data analytics service or the localization service, used to retrieve information about network performance or UE/vessel position, respectively.

In T&L sector, several EdgeApps interact with hardware devices deployed on field, such as IoT sensors, actuators, gateways, cameras, and Automated Guided Vehicles (AGVs). In the EdgeApp model, this is expressed as hardware dependency (the grey box on the left of Fig. 5.3), since the EdgeApp functionalities are strictly related to the interaction with these components, and the EdgeApp validation requires presence of these hardware devices in the testing environment. The service interfaces associated to each endpoint are also specified in terms of protocol and message format, and documented with protocol-specific interface specification (e.g., OpenAPI for REST APIs, SQL schemas, etc.) embedded in the EdgeApp package.

Such an abstract EdgeApp model has been designed to offer a service-oriented description of the EdgeApps, and to facilitate the verticals in the selection and composition of EdgeApps towards creating new vertical services for various use cases they want to build and test. Following this abstraction level captured in the EdgeApp blueprint and package, the vertical does not need to i) understand the details of the application internal structure, ii) know the deployment specifics over a virtualized infrastructure, or iii) understand the complex configuration of a 5G network slice. The orchestration-oriented and network-oriented model, captured by the VNF descriptor/package and any related 5G network slice template, remains hidden for the vertical and it is instead handled internally by the VITAL-5G platform for provisioning, lifecycle management, and testing purposes.

5.1.2 EdgeApps Classification

In order to make the concept of EdgeApps more palatable, and to assist users in the correct deployment, configuration and use of the appropriate EdgeApps for their specific use case, we have adopted a twofold classification of EdgeApps depending on i) their intended service, and ii) their composition and deployment methodology. This classification is presented also in the Table 5.1.

In terms of intended service, we categorize EdgeApps into vertical-specific and vertical-

agnostic.

- *Vertical-specific EdgeApps* implement functionalities designed specifically for a given vertical scenario and vertical service (as illustrated in Fig. 5.1). Its usage is related to a specific use case, and it is not designed to be easily customized or configured to adapt to different environments or services. Such EdgeApps provide strong focused services addressing specific and complex issues in a vertical domain (requires strong field expertise).
- *Vertical-agnostic EdgeApps* are generalized EdgeApps that can be easily adopted in different services since they support multiple customizations, data models and processing types. Examples of such EdgeApps are databases, message brokers, generalized monitoring probes or generalized AI/ML engines, as illustrated in Fig. 5.1.

In terms of composition, we make a differentiation between component-based and service-based EdgeApps, as follows:

- *Component-based EdgeApps* are atomic and elementary software components that operate in a generalized manner, and can be composed together and configured in a flexible manner, so that they can be customized to serve different purposes. A component-based EdgeApp can be delivered as a packaged set of software images with predefined configurations and interfaces, which can be then updated and customized to build more specialized EdgeApps delivering their own service.
- *Service-based EdgeApps* provide independent services that can be accessed in a standardized manner to support specific business requirements, and as such, they can be deployed in a stand-alone mode, without any dependency on additional EdgeApps, providing their own complete set of functionalities.

In general, multiple component-based EdgeApps, properly configured and customized, can be combined together to form a service-based EdgeApp.

5.2 EdgeApps for 5G T&L Vertical Services

The T&L sector is a major component of modern production and distributed systems, as it significantly contributes to the macroeconomic development [158]. However, processes in the T&L industry suffer from insufficient automation and optimization, which highly affects efficiency and safety of the T&L operations. We discuss these issues further in the context of a very specific example, such as T&L operations in the river/sea ports. In particular, Aroca et al. and Oliskevych et al. [159, 160] show that a highly specialized personnel in T&L industry (e.g., vessels captains, pilots, equipment, or train operators) is idle between 15-50% of the time, being conditioned by the availability of their assigned equipment. As the operational activities such as loading/unloading, and cruising on auto-pilot, do not require any intervention, this means that personnel can be engaged more efficiently with the help of new network capabilities, thereby including them in the remote operation of equipment as

well. On the other hand, a fast data transfer is a promising factor for ensuring safer T&L operations [161, 162]. To this end, a relatively high number of devices (e.g., sensors) needs to be connected to decision-making entities towards increasing safety of the port and logistics operation, by e.g., preventing equipment collisions in autonomous navigation, reacting to weather changes in advance, or identifying unexpected movements of other non-autonomous steered equipment.

Therefore, as Fig. 5.1 illustrates, it is expected that T&L industry leverages benefits brought by 5G technology, by integrating 5G ecosystem components into the infrastructure (e.g., ports, vessels, warehouses), and by developing 5G T&L services that experience enhanced KPIs through the use of uRLLC, eMBB, and mMTC, network slices.

Considering the study provided by Marquez-Barja et al. in [153], automated control of barges/vessels/ships requires bandwidth of 5-25 Mbps in the uplink, and latency lower than 22 ms, per High-definition (HD) video camera stream, and latency lower than 35 ms for vessel control interface. One example of such vertical service is illustrated in Fig. 5.1, where data-enabled assisted navigation of vessels in the river/sea ports needs an efficient data collection from sensors and cameras on the remote vessels. With the increase in number of connected vessels in the large ports, this requirement for uplink bandwidth becomes even more stringent (eMBB slice). Similarly, the remote crane control (Fig. 5.1) requires an end-to-end latency lower than 35 ms so that the remote operation can be performed efficiently and safely (uRLLC slice). Also, large and important ports are characterized by significant load, and require a more accurate control of the vessels, since the traffic is higher (i.e., low latency communication is needed), and the reliability is even more important.

Given the above-mentioned requirements, it is evident that only 5G technology is capable of providing faster and safer port operations with ms-level end-to-end latency, data rates of up to 20 Gbps, which are not available in 4G systems, as well as a stable, remote and real-time control. However, given the heterogeneity of data sources, edge, and cloud components, as well as scarcity of resources (edge infrastructure usually contains small amount of computing resources), an efficient resource management is needed to define the way vertical services are developed, deployed, and managed, on the 5G infrastructure.

In the scope of the VITAL-5G project, we have defined several use cases that aim to demonstrate the applicability of 5G EdgeApps to the realistic vertical service deployments that enable process automation and optimization, resource usage optimization, as well as improvements of time/cost efficiency [156]. In this section, we focus on the 5G-based river port use case, which leverages 5G connectivity and functionalities deployed as EdgeApps to improve performance and safety of the operations in a realistic environment such as Galati port. We first introduce the use case in Section 5.2.1, and then define the vertical services that are required for use case realization in the trial, as well as the EdgeApps that are building the identified vertical services.

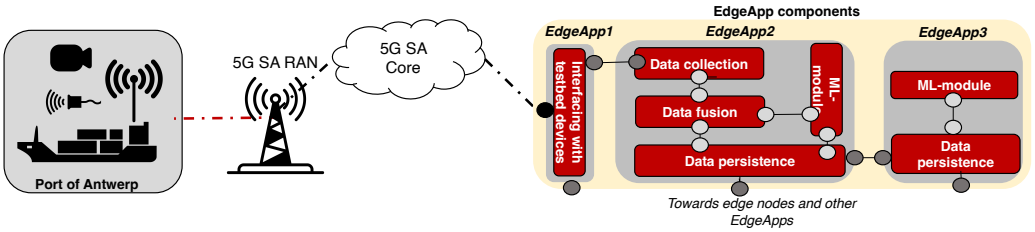


Figure 5.4: The representation of vertical services and EdgeApps described in Section 5.2.1.

5.2.1 Relevant use case: 5G connectivity and data-enabled assisted navigation using IoT sensing and video cameras

The use case is focused on the implementation of a data-enabled assisted navigation application using 5G network infrastructure, IoT sensing system and video cameras, as well as the ships and barges (cargos), which are altogether part of the Galati port in Romania. The Galati port is an entry point for large shipping traffic from the Black Sea towards the continental Europe, and is a part of the Rhine-Danube TEN-T Corridor[2]. It is the largest port on the Danube River, and the second largest Romanian port.

To realize the data-enabled assisted navigation use case, we design and propose two vertical service applications that enable a safer port operation of navigating ships by providing an operation/navigation assistance, even in severe weather and water conditions. Such vertical service will be performed on a series of ships belonging to the Romanian river transport company Navrom[3], which carries millions of tons of various goods through both internal and external routes towards the Western Europe.

The main objectives of this use case are:

- To reduce the number of dangerous navigation events (e.g., vessel collisions, or ships stuck in the river because of sandbanks or shallow waters) by collecting and transmitting the sensor and video data to the control units that optimize port operations.
- To reduce the logistics costs due to proper decisions based on an on-board diagnosis and monitoring functions, therefore limiting the impact of the human factor to take potentially wrong decisions.
- To create a more accurate electronic navigation map.

To achieve the above-mentioned objectives, it is essential to implement technologies for communication and monitoring of voyages in the activity of operating the ships. Thus, to avoid stationary downtime due to navigation errors, i.e., to reduce as much as possible the transport of empty units by achieving a higher percentage of loading, it is important to establish a better communication between ships and dispatchers. This can be achieved by

[2]Rhine-Danube TEN-T Corridor: The main east-west link across Continental Europe: https://transport.ec.europa.eu/transport-themes/infrastructure-and-investment/trans-european-transport-network-ten-t/rhine-danube-corridor_en

[3]Navrom: https://www.navrom.ro/index.php/ro/

enabling a real-time connectivity between the sensors that monitor the operating parameters of the ship, and the dispatcher office/navigation department. Also, to achieve higher levels of safety in sailing, a connection between the decision departments (e.g., the Fleet operation department) and ships is necessary for enabling assisted navigation that would handle difficult situations.

Concerning the connectivity, all sensors and cameras installed on the ships adopt the interoperable wireless protocols over a private 5G network, and enable the extension of Internet connectivity of the sensing system [62]. Several sensors (e.g., GPS, humidity, smoke, and engine power sensors) need to be installed on the ships and barges to collect relevant data, such as velocity, heading, and water/wind speed. Thanks to the 5G high-bandwidth and low-latency communication link, the EdgeApps presented in Section 5.2.2, fuse the live high-resolution video streams from the surroundings with the sensor data. With such an increased perception about the port, EdgeApps are coupled with an AI/ML module, producing relevant control signals for the captain and crew to take proper evidence-based decisions and provide an on-board diagnosis and predictive maintenance.

5.2.2 Related vertical-specific and vertical agnostic EdgeApps

In Fig. 5.4, we illustrate the use case described in Section 5.2.1, and the main EdgeApps that build the vertical services developed for such a use case. Both vertical services and corresponding EdgeApps are listed in Table 5.2. For our specific use case, we define two vertical services as follows:

- *Vertical service 1: Accurate electronic navigation maps creation* used for estimating the correct safe distance for a ship by using distributed sensor data ingestion,fusion, and post-processing. The data contains velocity, heading, water/wind speed, Global Navigation Satellite System (GNSS) data.
- *Vertical service 2: Predictive maintenance and sanity checks* applied on the sensor data for ship safety purposes, thereby using monitoring and on-board diagnostics data for limiting human error and potentially wrong decisions.

Following the VITAL-5G approach, these vertical services can be built as the composition of vertical-specific and vertical-agnostic EdgeApps, and in Sections 5.2.2, 5.2.2, and 5.2.2, we describe those EdgeApps as service building blocks.

EdgeApp1 - On board data collection and interfacing for a river vessel The *EdgeApp1* is a vertical-specific EdgeApp that collects data from i) on board sensors (i.e., water speed, water depth, outside/inside temperature, engine functional parameters, etc.), and ii) from video cameras placed on the river vessels. All data is made available for the local on-board server and provides the interface to/from external edge nodes and the 5G network. The data is formatted in a way to be understood and treated by EdgeApp2. Based on the input data, this EdgeApp exposes output through the 5G-based API endpoints on the edge nodes towards *EdgeApp2* and *EdgeApp3*. The output ranges from vertical-specific sensor data

Table 5.2: Vertical services and EdgeApps for the use described in Section 3.

Vertical service	EdgeApp
Accurate electronic navigation maps creation	EdgeApp1
	EdgeApp2
Predictive maintenance and sanity checks	EdgeApp1
	EdgeApp3

(such as water depth, and environmental parameters) to 5G infrastructure-related metrics (such as latency, availability, and uplink data rate). The *EdgeApp1* targets the second objective i.e., the cost reduction.

EdgeApp2 - Distributed sensor data ingestion, fusion and post-processing This vertical-agnostic EdgeApp is responsible for the ingestion of data from multiple distributed data sources, enhanced by data fusion and AI/ML-based analytics functionalities. The output that *EdgeApp2* produces supports reporting, advanced analytics, warnings (e.g. based on forecasts), and decisions, which all can be used by other EdgeApps. In particular, *EdgeApp2* is responsible for the following:

- Ingestion of data from various sources (including the 5G infrastructure, as well as the T&L devices) to enable training and testing of AI/ML models. The data collection component performs acquisition of both streaming and batch data, through the use of HTTP REST APIs, thus querying the data source directly or through the message bus services.
- Transformation of datasets is performed by the data fusion component. It offers capabilities for various logical transformations of data, such as cleaning and inserting missing values, time-space correlation, and transformations of unstructured datasets to structured and vice versa.
- Post-processing of data, starting from simple functionalities like aggregation and ranging to applying AI/ML models that enable advanced analytics. The AI/ML module is responsible for the training and deployment of post-processing procedures on the fused data. This module consists of i) submodules that apply unsupervised and supervised learning, the autotune engine, and the AI/ML registry. The autotune engine ensures the automated calibration of multiple supervised or unsupervised models that are trained in submodules, and then saved in the AI/ML registry.
- Data persistence for the data to be further utilized by other EdgeApps is performed by the data persistence component, which enables i) the storage of both structured and unstructured data, whether raw, or fused and post-processed as the result of the AI/ML models, and ii) exposing data through the appropriate APIs.
- Corresponding interfaces (APIs) for other components (e.g. dashboard), other EdgeApps, and 3rd parties, to utilize the enhanced information that has been produced and stored.

EdgeApp3 – Predictive maintenance This vertical agnostic EdgeApp utilizes data exposed by *EdgeApp2*, but it can consume input data from various sources. The output is the results of advanced AI/ML-based diagnostics. The *EdgeApp3* is responsible for the deployment of supervised and/or unsupervised modelling techniques to achieve functions like automated labelling, outlier detection, trace back analysis, graphical representations, predictions for the near future, and therefore support decision-making for predictive maintenance. In particular, data is fed to an AI/ML module consisting of various sub-components, which enables the automated model usage and calibration of multiple AI/ML models, as well as the extraction of meaningful results. The results are then sent to a data persistence component, which is also responsible for the exposure of the results to other components/EdgeApps through the appropriate interfaces.

5.3 EdgeApps for enhancing back-situation awareness in automotive services

In this section, we steer the focus to the automotive sector, and show one particular type of vertical service, i.e., back-situation awareness on the highways, which is leveraging the concept of EdgeApps to increase the awareness of emergency vehicles on the roads. MEC and NFV are considered as one of the key technology enablers for 5G and beyond [163], and MEC systems especially are leveraged for empowering applications with 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 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 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. 5.5. 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. 5.5). 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.

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) [164] towards the newly instantiated BSA application on the MEC systems (see red arrow line in Fig. 5.5), for EmV's each Global Positioning

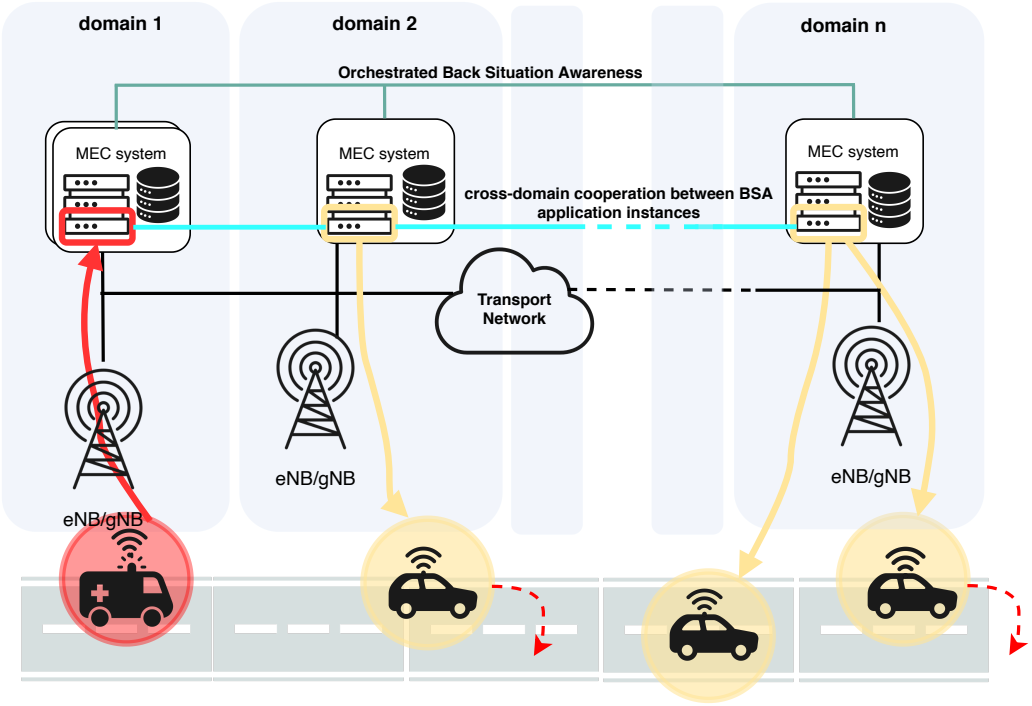


Figure 5.5: Back situation awareness on the highways.

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 dissemination areas, which the BSA application derives along the route-path. The computed ETA values are then encoded in the C-ITS Decentralized Environmental Notification Messages (DENMs) [165], which are broadcasted in the geographic regions bound to dissemination areas relevant to the encoded ETA value (see yellow arrow line in Fig. 5.5). In the following we denote the distribution of DENMs in the dissemination areas as geocast.

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 in which they are running, as they can proactively inform orchestration entities about the need for an application instance in the other domain, as well as of ii) the other peering applications from the other edge domains to which they need to connect.

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 [72, 71, 86], and how such 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 V2V communication that sends the required information about an EmV only in a close vicinity from this EmV [77, 79], thereby only addressing those vehicles that are approximately 300m away from them [166]. 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. For example, the observational cohort study with 10,315 cases transported by four English ambulance services [86] 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 [80], who propose a V2I approach, but this is still not enough given that transmission range of roadside units is between 400 and 500m, with the average delay in message transmission of 487 ms and 574 ms [167].

To address the aforementioned gaps in existing approaches, our BSA system relies on the V2N communication, i.e., 5G-based MEC deployment where BSA application is running on the optimally selected edge cloud. Given that information such as 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. 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 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 [85], assessing the ETA accuracy and error estimation. In this paper, we focus on the overall BSA system, 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[4] testbed. This paper also introduces a new 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

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

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. Thus, 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 this section is organized as follows. Section 5.3.1 presents the system design, the BSA service architecture, and the multi-domain aspects of BSA operation. This is followed by Section 5.3.2 providing detailed performance results, and analysis and discussion, based on an experimental setup.

5.3.1 BSA application - System Design and Architecture

In this section, we provide an overview of all 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 MEC application for a V2X use case addressing emergency situations on the roads, and later in Section 5.3.2 to evaluate its performance in a realistic environment. Here we also detail on the operational aspects of the BSA application, stretching multiple domains while being orchestrated by an optimized MEC orchestration system.

5.3.1.1 BSA Application Overview

Fig. 5.6 shows the system overview of the BSA application in the context of the standardized ETSI MEC system architecture [50], and also attempts to depict it in the context of the use case depicted in Fig. 5.5. It should be noted that Fig. 5.6 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. 5.6, 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 points 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 [164] received periodically from the EmV's with reference to these Waypoints (WPs). Such mechanism of proactive notifications allows the drivers to deduce the maneuver recommendations. The detailed logic, including the analytical model and the performance evaluation of the ETA algorithm, have been described and analyzed in [85] in terms of error in the estimation of ETA values.

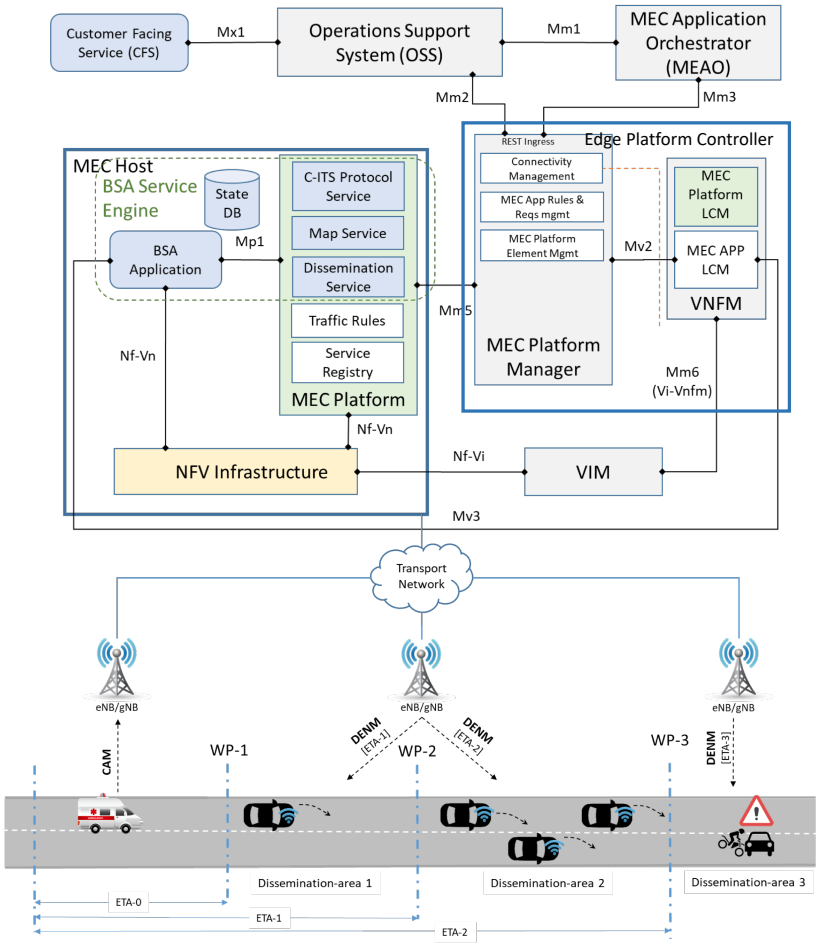


Figure 5.6: Overview of the BSA service system design.

2. **C-ITS Protocol Service:** This is proposed to be a microservice for decoding/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[5] and the decoding/encoding helper function entities in Fig. 5.7, which will be explained in 5.3.1.2.

3. **Map Service:** This is proposed to be a microservice that can be consumed by the ETA algorithm for getting geo-spatial 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 a storage where the meta-data/state-

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

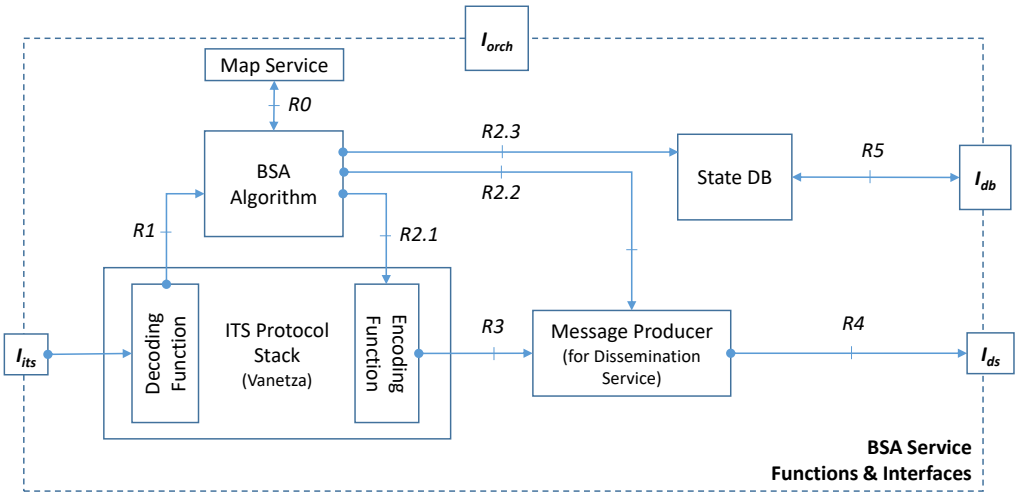


Figure 5.7: Back Situation Awareness (BSA) Service Architecture - Functional Elements and Interfaces.

information of the EmV decoded/parsed by the C-ITS protocol service from the periodically received CAMs/DENMs 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. 5.6.

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 [165], which is then geo-casted in the respective dissemination area with the help of the dissemination service. For example, as depicted in Fig. 5.6, 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.

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. 5.6 are specified in the ETSI GS MEC 003 v2.1.1 standard [50] 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 is out of scope of this paper.

5.3.1.2 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 defacto choice for Communication Service Providers (CSPs). This approach allows CSPs to deploy services rapidly and flexibly, with reduced Capital Expenditure (CapEx) and Operational 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 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 and uplink bandwidth for V2X applications [168], 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 5.3.1.1), 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 5.3.1.1 is achieved for the BSA application.

In Fig. 5.7, 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. 5.6, 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 towards 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. 5.7).
- 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.

5.3.1.3 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. 5.5, while Fig. 5.6 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 a long corridor on the highway, and as such, it requires a proper management and orchestration to achieve a smooth cross-domain operation. Thus, in Fig. 5.8, 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 termination of application instances that are not used by the EmV anymore, thereby releasing MEC resources for other types of services.

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 5.3.1.1, 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. 5.8). 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-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 affected by emergency situation) 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 management and orchestration of BSA application service running in all domains simultaneously (steps 2-4, Fig. 5.8). 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 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. 5.8). In particular, Fig. 5.8 hides the complexity of the MEC orchestration system, but Fig. 5.6 shows that it comprises as key elements an Edge Orchestration component, i.e., MEAO, and an Edge Platform Controller [142]. More details on the orchestration elements and operations are described in our work that studied collaborative orchestration for V2X services [169], explaining that the Edge Platform Controller extends the open-source container orchestration platform Kubernetes[6] 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[7]. Thus, the Edge Platform Controller enforces the tasks such as 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. 5.8), and to disseminate C-ITS DENMs to other vehicles, as described in Section 5.3.1.2.

Phase 2: Dynamic deployment and runtime of peering application instances The instantiation can be performed simultaneously on multiple edges per coordination 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 platform can remain application-agnostic, allowing applications themselves

[6]Kubernetes Project Portal: `https://kubernetes.io/`

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

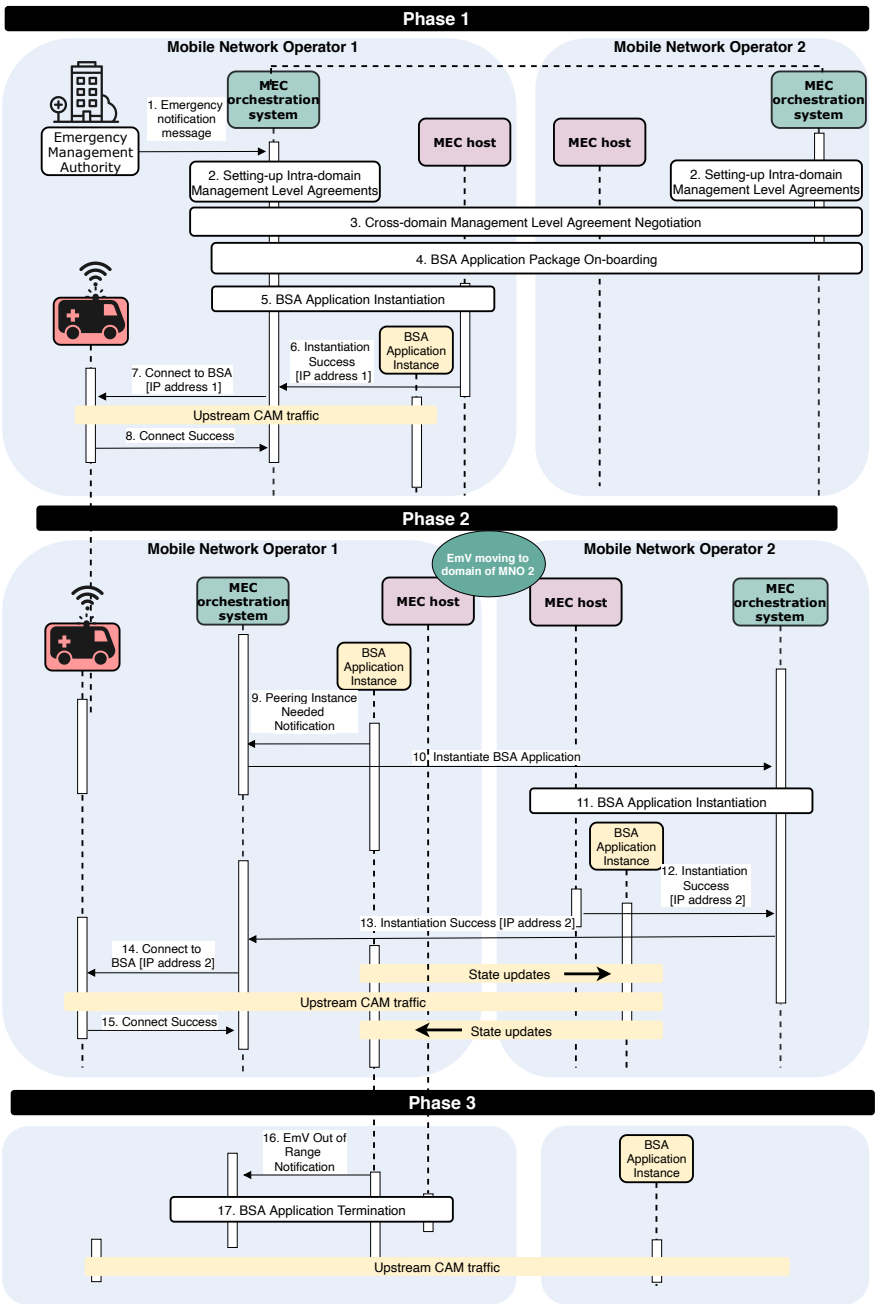


Figure 5.8: Overview of multi-domain operations of the federated multi-domain BSA service; Yellow boxes imply the operations that are contributors to the KPIs we measured (e.g., upstream CAM traffic affected by communication latency, BSA application instance producing computational latency and CPU/RAM load, and state updates over network effected by state update delay).

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

Table 5.3: 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	32 GB
Processor	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz
Storage	C610/X99 series chipset sSATA Controller [AHCI mode]
Disk	1TB Samsung SSD 860
Network	I350 Gigabit Network Connection

second. Thus, in Phase 2 depicted in Fig. 5.8, 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. 5.8). 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. 5.8.

Phase 3: Dynamic termination of application instances During the application runtime, the orchestration entities make sure that application instances have sufficient amount of resources to perform required operations (e.g., by performing scaling operations). Once the resources are not needed, such as in the Phase 3 in domain 1 (steps 16-17, Fig. 5.8), 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.

5.3.2 Performance evaluation

In this section we present the performance evaluation of 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, ii) defining the set of metrics that reflect service performance, and iii) providing the evaluation results.

5.3.2.1 Smart Highway testbed setup for experimentatal evaluation

In order to conduct the experimental evaluation of our BSA application in a realistic environment, we deployed application instances on top of the MEC hosts within an orchestrated vehicular system, i.e., the Smart Highway testbed. The Smart Highway testbed [170] 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. 5.9 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 5.3.

As this research is conducted in the context of the 5G-CARMEN project, which is focused on leveraging 5G advancements to deliver a safer and more intelligent transportation on the Bologna-Munich corridor, we designed our experiment on the Smart Highway to stretch multiple domains, i.e., including the border 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 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 on-board unit of the physical vehicle we utilized in the Smart Highway setup (Fig. 5.9). This service, although running on a physical on-board unit of a testbed vehicle, is emulating the movement on the corridor between Italy and Austria. 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 30m/s, which is in the range of the speed limit for the European highways. All CAM messages carrying a real-time information on the EmV 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.

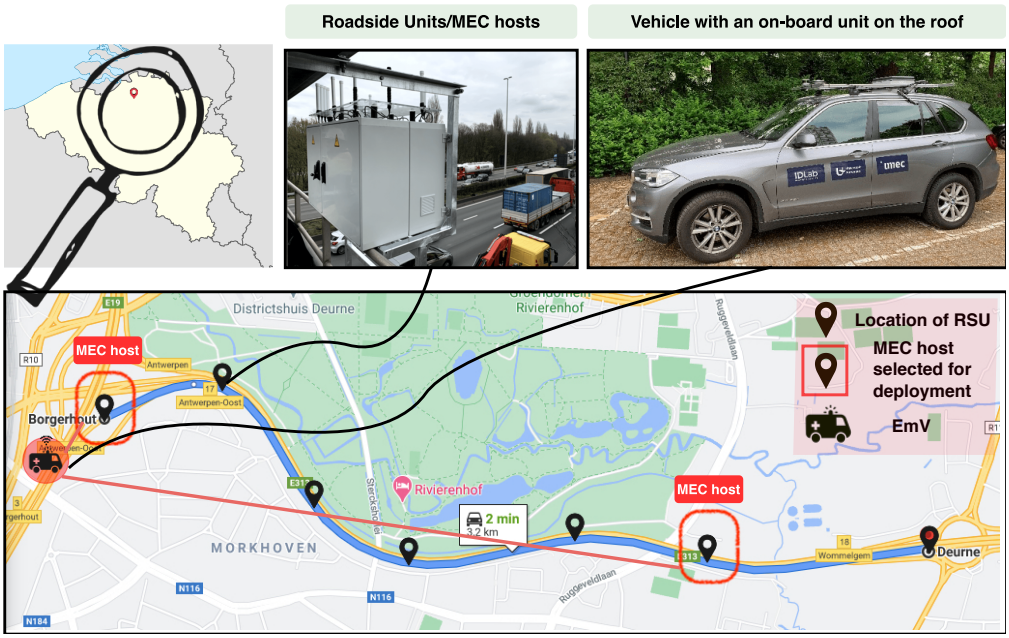


Figure 5.9: The experimental setup consisting of the vehicle, and the MEC hosts deployed on the E313 highway (Antwerp, Belgium).

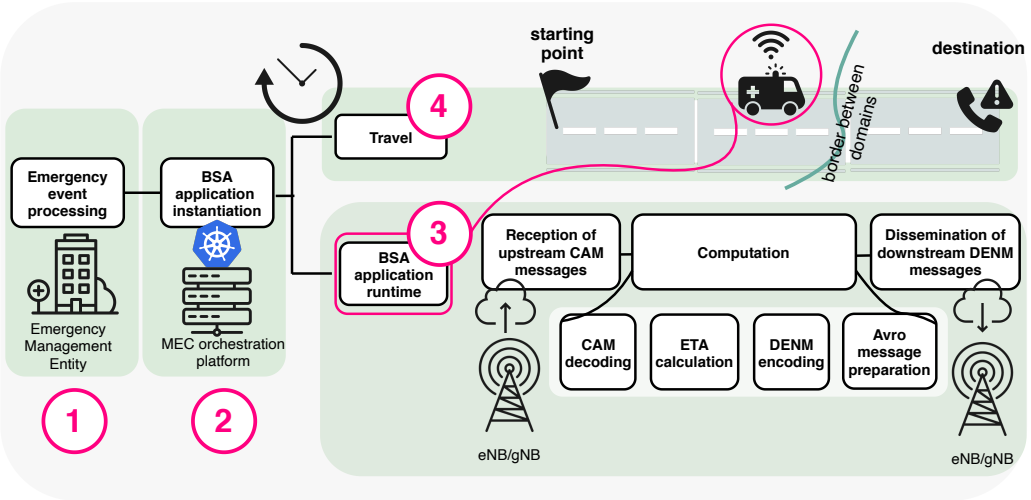


Figure 5.10: The overall emergency response time in the BSA system.

5.3.2.2 Key Performance Indicators

The main goal of deploying BSA application service on top of the orchestrated MEC systems is improving 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 emergency scale on the MEC system resources and service response

time. In Fig. 5.10, 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. 5.10), ii) application instantiation on top of the orchestrated MEC system (contributor 2), iii) MEC application runtime while EmV is travelling (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 5.3.1.3, 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 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 5.3.1.3, and after decision is made by orchestrators, the Edge Platform controller [169] 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. 5.10, 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. 5.11. 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. 5.11), for deriving ETA values for respective dissemination area (i.e., T_{comp} in Fig. 5.11). The derived ETA values are encoded inside the DENMs (i.e., T_{encode} in Fig. 5.11), which are generated for the respective dissemination areas to notify the civilian vehicles, and to prepare the required format[8] for the message dissemination service on the MEC system. All this accounts towards the *computational delay*. Fig. 5.11 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. 5.11. 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

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

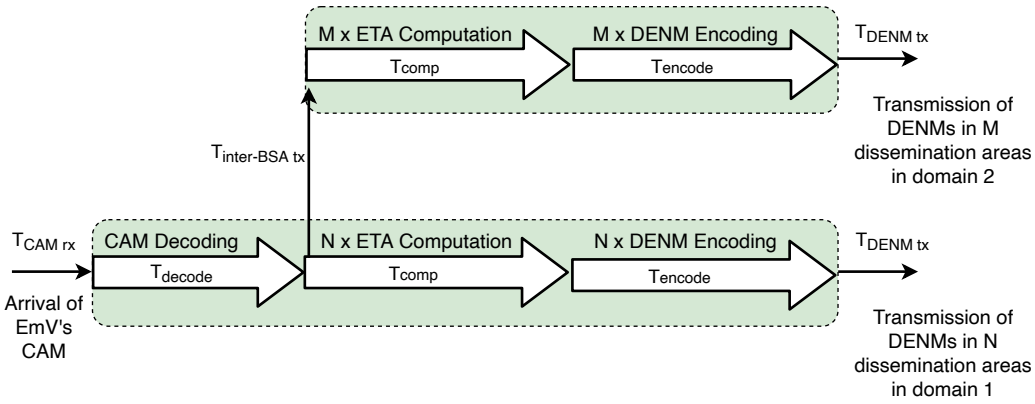


Figure 5.11: Visualization of contributors to the overall BSA response time.

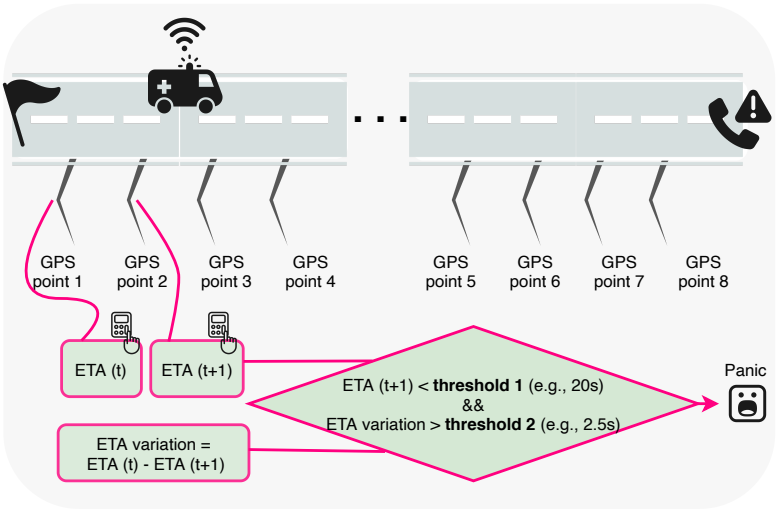


Figure 5.12: Panic indicator evaluation per dissemination area.

the j-th EmV ($j \in \{1, 2, ..., v\}$). Concerning the overall communication latency, described as T_{comm_i} in equation (5.1), 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 [145, 146, 169], which are described further in (5.1), 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 our performance evaluation scenario, the amount of data that is being transferred in each CAM (with frequency f_{CAM}) and DENM message is 189 B (i.e., $x_i = 189$ B), containing the usual C-ITS headers and payload carrying the relevant information for the BSA application (e.g., speed/heading/location of EmV, ETA

values).

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, which is bounded by the speed of light in a vacuum. The parameters β and γ are defined as weighting factors that balance the networking characteristics [146, 169], and as described in Chapter 4, we can define β as equal to 1 if bandwidth is considered stable, while γ is a refraction index for the medium other than a vacuum through which the electromagnetic signals are traversing. In our scenario, signals carrying CAM and DENM messages from/to emergency vehicle to/from BSA application service running at the MEC are being transmitted via 5G Uu link, i.e., via wireless link for which s is usually considered as equal to 300000 km/s in the literature [145, 146, 169]. Due to the close proximity of the edge nodes from the vehicles on the road (l_{ij} less than 10km), the propagation latency is negligible in this case, as it is less than 1 ms.

$$\begin{aligned} T_{comm_i} = T_{t_{ij}} + T_{p_{ij}} &= T_{CAMrx_i} + T_{DENMtx_i} \\ T_{t_{ij}} &= \beta \cdot \frac{x_i}{B_{ij}} \\ T_{p_{ij}} &= \gamma \cdot \frac{l_{ij}}{s} \end{aligned} \tag{5.1}$$

On the other hand, for the computational latency illustrated in Fig. 5.11 and described as T_{comp_i} in equation (5.4), 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.

Let us study this type of latency a bit more. If a MEC system is considered as a model where CAM messages are arriving as an $M|M|k$ queue model [171], the occupation of the processor on the MEC host can be defined as ρ in 5.2, 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.

$$\rho = \frac{f_{CAM}}{k \cdot \mu} \tag{5.2}$$

$$P_w = \frac{(k \cdot \rho)^k}{k!} \cdot \left((1-\rho) \cdot \sum_{n=0}^{k-1} \frac{(k \cdot \rho)^n}{n!} + \frac{(k \cdot \rho)^k}{k!} \right)^{-1} \tag{5.3}$$

According to [171], the probability that a certain task needs to wait to be processed is described as P_w in (5.3), which further contributes to the definition of the computational latency in (5.4), consisting of the wait time and service time.

$$T_{comp_i} = T_{w_i} + T_{s_i} = P_{w_i} \cdot (1-\rho_i)^{-1} \cdot (k_i \cdot \rho_i)^{-1} + \frac{1}{\mu_i} \tag{5.4}$$

$$T_{comp_i} = T_{decode_i} + T_{eta_i} + T_{encode_i} \tag{5.5}$$

To study the impact that a 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 (5.6), 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 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, 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 cooperative maneuvering 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).

$$T_{comp_i} \leq \frac{1}{f_{CAM}} \tag{5.6}$$

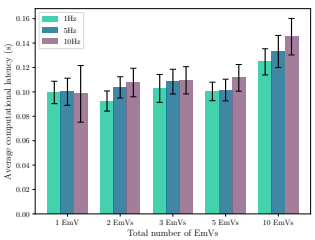
Another important metric is the state update delay T_{sud} defined in (5.7), which is specific for multi-domain deployments where multiple peering BSA application services are running on different MEC hosts, while addressing the same emergency situation in a distributed way. This metric is equivalent to communication latency described in (5.1), 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 $l_{m_1 m_2}$. In our scenario, the same rationale described for equation (5.1) applies to β and γ here as well, while x_{meta} is approximately 150 B (data exchanged between two BSA instances running on two edges, informing each other about 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.

$$T_{sud} = \beta \cdot \frac{x_{meta}}{B_{m_1 m_2}} + \gamma \cdot \frac{l_{m_1 m_2}}{s} \tag{5.7}$$

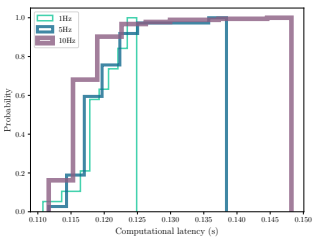
The reason we defined the model of both computational and communication latency is to better understand the experimentation results presented in Section 5.3.2.3, and to grasp the role of all contributing factors in the latencies we achieved during the experimentation analysis.

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 number of vehicles that consume BSA service simultaneously.

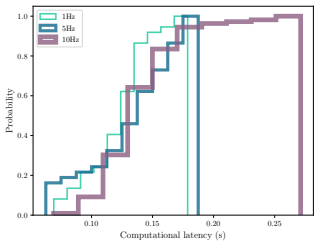
Finally, we introduce a metric called *panic indicator* to depict how our BSA application service can potentially help civilian vehicles to clear the lane for the EmV in a calm manner, thereby increasing the road safety. In Fig. 5.12, we showcase how panic indicator can be calculated for each dissemination 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.,



(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.

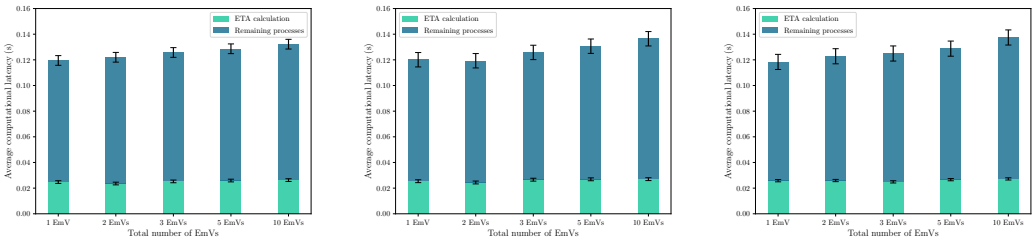
Figure 5.13: The overall computational latency with reference to single EmV.

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. 5.12). This threshold determines whether the difference between two successive ETA values can cause panic, such as 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 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. 5.12), as it indicates whether EmV is approaching in a short time frame or not. In case that 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 is not met, indicator is equal to zero. It is important to note that the 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 [172], 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.

5.3.2.3 Results and discussion

Computational and communication latency As described in Section 5.3.2.2, BSA application service can receive CAMs from EmVs with different frequencies, and accordingly, in Fig. 5.13 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. 5.13a 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 number of vehicles, a more visible



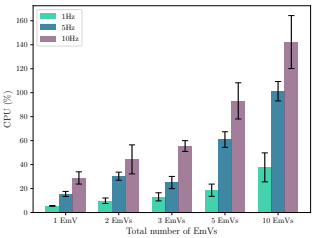
(a) CAM frequency 1 Hz. *(b) CAM frequency 5 Hz.* *(c) CAM frequency 10 Hz.*

Figure 5.14: The overall computational latency per process.

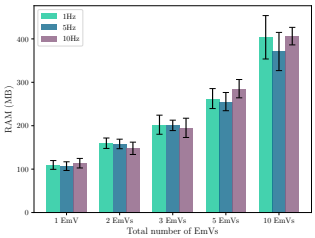
difference between cases can be seen in Figures 5.13b and 5.13c that show the Cumulative Distribution Function (CDF) of latency with reference to single EmV, in case there is one EmV, and ten EmVs (i.e., large scale emergency), respectively.

From the results presented in Figures 5.13b and 5.13c, 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. 5.13b, 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. 5.13c), 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. 5.13b). In the same case the probability drops to 0.65 if BSA service is serving 10 EmVs simultaneously (Fig. 5.13c), meaning that there is even a 35% chance that computational latency is above 150ms, which leads to 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 (5.6)) is not sufficient for the service to perform all operations illustrated in Fig. 5.10, 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 other two frequencies.

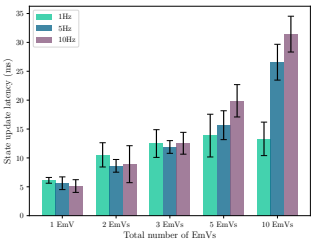
If we now take a look at the average computational latency per specific BSA operation (Fig. 5.14), it can be seen that in all cases less than 20% of the overall computational delay is incurred by algorithm that evaluates ETA for all dissemination areas, based on the latest data on the speed/latency. 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 case of scaling the whole single container BSA application.



(a) Average CPU consumption.



(b) Average RAM consumption.

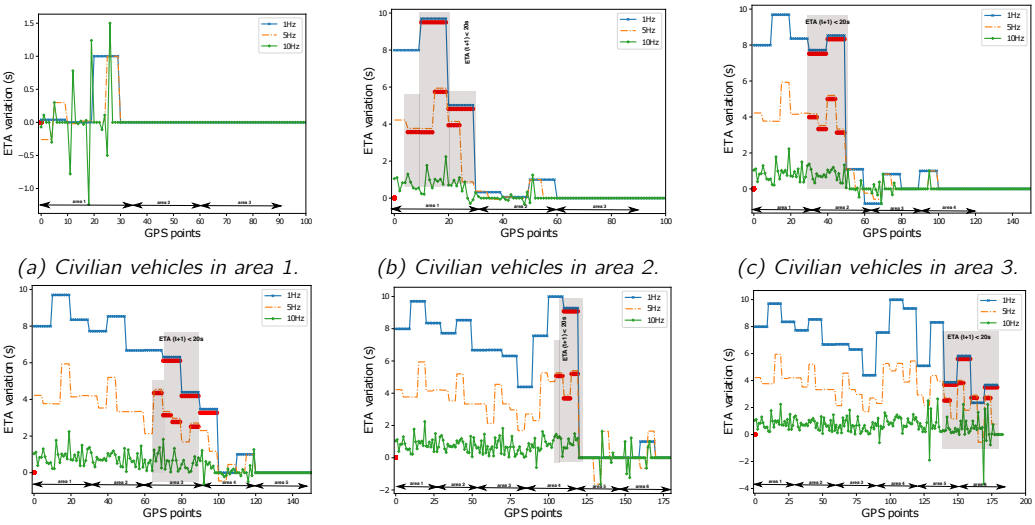


(c) Average state update latency.

Figure 5.15: The resource consumption and state update latency.

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 case of 1 Hz, the time frame for computation is 1 s, which is sufficient, according to the results presented in Fig. 5.13. However, if frequency is 10 Hz, 100 ms seem 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 accuracy of ETA calculation at the same time. This way, although not configuring the CAM generation frequency at the vehicle side, 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 5.3.1.3, 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, 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. 5.15c. 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. 5.15c, 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. 5.13a, 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 application from one edge to another. The statistical test also supports previously presented result, showing that the state update latency significantly changes with the CAM frequency



(a) Civilian vehicles in area 1. *(b) Civilian vehicles in area 2.* *(c) Civilian vehicles in area 3.*

(d) Civilian vehicles in area 4. *(e) Civilian vehicles in area 5.* *(f) Civilian vehicles in area 6.*

Figure 5.16: ETA Variation ($ETA(t) - ETA(t+1)$); The values highlighted in red implying that panic indicator is on, for civilian vehicles in dissemination areas 1 to 6.

($p_{value} = 2.2845e-05$), 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. 5.15c), the state update duration might affect the accuracy of calculating the ETA values for dissemination areas, it needs to be accounted in ETA algorithm.

Concerning the communication latency described in Section 5.3.2.2, 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 5.3.2.1. 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 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 the average uplink latency of 28.84ms, with the standard deviation of 18.64ms, and the average downlink latency 18.63, and the standard deviation of 6.39ms.

Measuring uplink latency for 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 the information such as current location of a vehicle, the longer uplink latency can significantly affect the efficiency of the V2X service. For example, if vehicle is driving with the speed of 100 km/h, the uplink latency of 50 ms will affect the quality of data, because

the vehicle will already move for an additional ca. 1.4 m until V2X service receives this data. In the case of autonomous vehicles, such delay is of course not tolerable, and that is why ultra-low latency promised by 5G and MEC is important. Concerning the BSA service, such delay can affect the accuracy of ETA algorithm, and thus, 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. When it comes to the downlink latency, less than 20 ms latency that we obtained and presented in Section 5.3.2.3 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 manoeuvre 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 in downlink, improvements in latency brought by 5G play a significant role for V2X services.

Resource consumption The increase in CAM frequency and number of EmV concurrently served not only increases the computational latency, but it also highly affects the resource consumption of the containerized application. As it can be seen in Fig. 5.15, 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.

Panic indicator As elaborated in Section 5.3.2.2, 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. 5.12), considering the route with six dissemination areas in total. We calculate ETA variation that would be experienced by civilian vehicles in these six dissemination areas, and present it in Fig. 5.16. According to the description of panic indicator as metric, which is provided in Section 5.3.2.2, we indicate that panic happens in situations in which the current ETA value for a specific area is less than 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 this criteria is met, 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 5.16a)-5.16f) show the ETA variation ($ETA(t) - ETA(t+1)$) for civilian vehicles in respective areas from 1 to 6. For example, in Fig. 5.16f, 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 x-axis). The portions of the graphs that are highlighted in grey represent the cases when ETA variation is larger than 2.5s, and the current ETA is lower than 20s (i.e., both criteria from Fig. 5.12 met). With reference to Fig. 5.16f, 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. 5.16), 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 frequency is 10 Hz, but from the results studied above, we clearly identified several bottlenecks of having such 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, in order to decrease the probability of panic, thereby improving their efficiency of responding to EmV's arrival.

The overall emergency response time If all vehicles on the highway are equipped with corresponding on-board units, thus, being able to receive notifications from BSA application via message dissemination service, the largest portion of the overall response time is mainly determined by the travel time of the EmV. Thus, for the route between Italy and Austria, which we considered for emulating the movement of the EmV on the testing highway, the maximum speed allowed is around 140 km/h, which results in approximately 2.07min of travel time. Assuming that due to the early notifications received by MEC application from BSA, all vehicles efficiently clear the lane for approaching EmV, the reduced travel time will be almost three times lower than the average time usually needed for a vehicle to reach this specific destination from the same starting point.

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 the 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 than it 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 5.3.1. 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., 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. Finally, if 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.

5.4 Summary of the Chapter

In this Chapter, we define the concept of EdgeApps with the goal to: i) abstract the complexity of 5G network and infrastructure configuration in providing vertical services tailored to automotive and T&L sectors, and ii) to facilitate service deployment in real-life environments (e.g., harbours and busy highways). The lifetime of EdgeApps is fluid, as they can be designed and created on-demand to boost specific aspects of safety and efficiency in vertical operations through the delivery of vertical services for e.g., assisted maneuvering, preventing equipment collisions, in-advance preparation for weather changes, and efficient reactions to emergency situations on the roads. Although 5G is providing the means for enhancing T&L operations through creating network slices, these slices need to be configured to service-specific needs. Therefore, EdgeApps are the glue that binds the requirements coming from the vertical services, and the actual service deployments using 5G network and virtualized infrastructure resources. This is achieved by extending the ETSI VNF concept to include relevant service-specific information, as well as mobile connectivity requirements (5G slice profile, and 5G core services) that are translated to 5G network slice profiles in the EdgeApp blueprint.

Furthermore, we provided two examples of how EdgeApps can be applied to different verticals, i.e., T&L and automotive. The T&L sector requires a constant improvement of safety and efficiency of operations, and EdgeApps are created to enable industrial stakeholders to more flexibly leverage 5G benefits, by mapping their user requirements to the actual 5G network slices that will improve the overall service quality. As the work on this thesis was more focused on the automotive use cases, in this Chapter we also put more emphasis on a particular type of EdgeApp, i.e., application service, for enhancing back situation awareness on the highways. Such a EdgeApp enables early notifications for vehicles about the ETA of an approaching EmV. Due to the significant importance of decreasing the overall response time to the emergency events, we performed a thorough performance analysis of the BSA application service, measuring the impact of emergency on the MEC system resources, and service response time. Moreover, we introduced a metric called *panic indicator* that provides a notion on how the proposed BSA EdgeApp 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. Thus, in this Chapter, we derived 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.

Finally, to realize the true potential of EdgeApps for any type of vertical, there is a need

for an efficient EdgeApp management and orchestration as presented in Chapter 4, which constantly is constantly monitoring the performance of EdgeApps, thereby deriving and enforcing decisions that will either maintain or improve overall service quality. After presenting such orchestration mechanisms in Chapter 4, and detailing on vertical service design deployment mechanisms through applying the concept of EdgeApps in Chapter 5, we dig deeper into AI-enhanced orchestration, and opportunities to further improve performance of orchestration systems by making them automated and intelligent.

Chapter 6

Mechanisms for intelligent and automated edge orchestration, and future of MANO

This chapter is part of the **Contribution 4**: *Intelligent and automated management and orchestration of services and resources*, and it is based on:

N. Slamnik-Krijestorac, M. Camelo, C. Y. Chang, P. Soto-Arenas, L. Cominardi, D. De Vleeschauwer, S. Latré, and J. Marquez-Barja, "AI-empowered Management and Orchestration of Vehicular Systems in the Beyond 5G era *(Submitted)*," in *IEEE Intelligent Transportation Systems Magazine*, pp. 1–7, 2022, Impact factor: 5.293.

N. Slamnik-Kriještorac, M. C. Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Building Realistic Experimentation Environments for AI-enhanced Management and Orchestration (MANO) of 5G and beyond V2X systems," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 437-440, doi: 10.1109/CCNC49033.2022.9700649.

N. Slamnik-Kriještorac, P. Soto-Arenas, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "Realistic Experimentation Environments for Intelligent and Distributed Management and Orchestration (MANO) in 5G and beyond," 2022 *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 943-944, doi: 10.1109/CCNC49033.2022.9700659.

N. Slamnik-Kriještorac, S. Latré and J. M. Marquez-Barja, "An optimized application-context relocation approach for Connected and Automated Mobility (CAM) ," *IEEE 5G for Connected and Automated Mobility (CAM)*, 2021, doi: 10.48550/arXiv.2109.11362.

N. Slamnik-Kriještorac, M. Camelo Botero, L. Cominardi, S. Latré and J. M. Marquez-Barja, "An ML-driven framework for edge orchestration in a vehicular NFV MANO environment *Accepted*," 2022 *IEEE 20th Annual Consumer Communications & Networking Conference (CCNC)*, 2023, Core ranking: B.

6.1 Toward Automated MANO

Due to extreme-low latency (1-10 ms), ultra-high reliability (99,999%), enhanced throughput (above 100 Mbps up to 20 Gbps), and flexible resource usage, the B5G ecosystem offers opportunities to verticals [151], e.g., automotive sector, as illustrated in Fig. 6.1, to improve the existing services and create new ones that were not feasible before. In previous Chapter, we discussed how these services can be deployed as EdgeApps in order to gain the full potential of 5G systems, and to facilitate their usage in the context of vertical industries. Some of these new V2X services that are delivered through EdgeApps are shown in Fig. 6.1, such as i) maneuver recommendation that instruct vehicles on the path/speed, ii) collision avoidance, iii) teleoperation supporting remotely-operated vehicles, and iv) infotainment (e.g., video streaming).

To realize such services, a cellular B5G system (i.e., radio access and core network) is used together with the managed and orchestrated infrastructure that provides distributed EdgeApps [169], spanning different edge domains that may belong to different MNOs (cf. Fig. 6.1). These EdgeApps are used as building blocks for new services, which are enabled by well-recognized pillars of B5G mobile communication systems such as SDN, NFV, and MEC.

To provide V2X services in a reliable and responsive manner by localizing access to virtualized network resources and services in the B5G ecosystem, challenges such as operating under constrained resources, with heterogeneous network edges, and in time-varying conditions, must be carefully addressed [169]. These challenges are particularly important in the highly mobile environments with connected vehicles, since V2X services require continuous monitoring of network and computing resources, and efficient and swift service control (e.g., fast scaling, redeployment, migration) and resource optimization, with reference to mobility patterns and resource demand [169]. Such an increasing set of control variables and optimization targets will make the B5G V2X system ultimately complex, whereas traditional MANO processes, which are either open-loop and inherently manual or closed-loop but slow (e.g., human-in-the-loop) and based on simple and static rules, need to be improved or even replaced with techniques that automate such MANO processes.

The potential of integrating AI/ML techniques with MANO processes is well recognized, and some research efforts have been devoted to this topic [173, 174, 96], focusing on enforcing and automating NFV MANO operations. The NFV MANO operations, e.g., service placement, scaling, and/or migration, can be performed to achieve service continuity by leveraging data analytics and AI/ML techniques for event anticipation, fast response, and advance preparation of network [169]. In particular, AI/ML can provide the NI for MANO systems through the NIFs, which are the pipelines of effective AI/ML algorithms that detect/anticipate new requests or fluctuations in the network activities [175], and help orchestrators to respond to such changes in a fully automated manner. However, there is still a lack of full understanding of the selection, deployment, and impact of AI/ML on the NFV MANO operations for V2X services.

To this end, in this Chapter, we first examine the gaps in current NFV MANO solutions for B5G-based V2X services and analyze the potential of bringing NIFs to NFV MANO, in the form of various AI/ML techniques that can bridge the identified gaps. Along with this

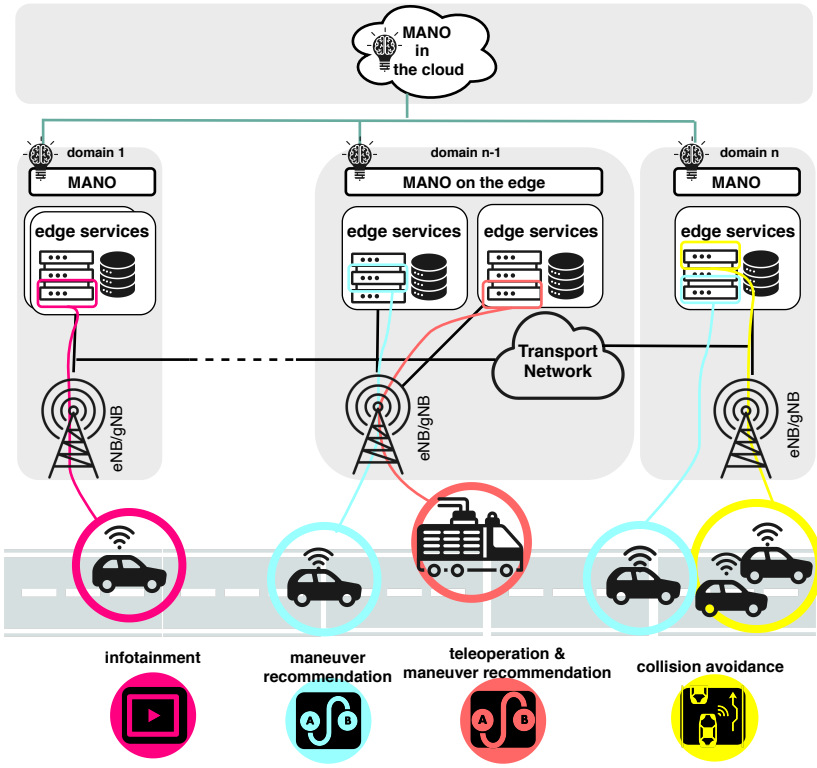


Figure 6.1: NFV MANO in B5G C-V2X system.

gap analysis, we study how and which AI/ML techniques have the potential to improve and automate the MANO operations, and what are the implications that need be further studied. Afterwards, in Section 6.2, we present our efforts on leveraging AIML towards enhancing edge orchestration, showcasing the results of performance evaluation conducted in a real-life environment.

6.1.1 Gaps in the current NFV MANO solutions

The NFV MANO systems perform service instantiation/placement, scaling, migration, and termination, based on information gathered from various network segments. By studying the existing solutions and their applicability to V2X service orchestration, we identify several gaps that need to be carefully addressed.

Manual orchestration operations The stringent requirements for V2X services, with self-driving vehicles as an ultimate goal, require extensive broadband (especially on uplink) [153], resilient and reliable connectivity, and network availability up to five-nines [176]. This urges for real-time monitoring of the network performance to achieve an improved decision-making.

The efficiency of NFV MANO operations needs to be improved (e.g., lengthy scaling procedure that hinders service reliability and response time), for two reasons: i) the operations of processing monitored data and making decisions are traditionally manual and require human intervention that is prone to mistakes and additional delays, and ii) network complexity significantly increases with heterogeneous and distributed resources and services [174], which is even more significant in V2X systems because of the presence of various automobile manufacturers, vehicle application providers, MEC service providers, and MNOs. Thus, AI/ML techniques have the potential to support the automation of NFV MANO operations by combining data analytics and learning in closed-loop control, thereby outperforming traditional optimization schemes that are complex and lengthy, heuristic ones that are problem-specific and domain-dependent, and open-loop approaches that are prone to human errors, which makes them all ineffective in swift responses to dynamic network changes [174].

KPI fluctuations Dynamic changes in KPIs occur due to fluctuations in the demands from vehicles, and their mobility patterns, which is particularly challenging when large numbers of moving vehicles are simultaneously connected to the orchestrated edge services. Thus, orchestrators need to improve their operation by learning from the environment, identifying or even predicting changes in KPIs, and translating these changes into required NFV MANO operations that will maintain service performance at the desired level.

Increased load of NFV MANO From 5G onward, both cloudification and virtualization concepts are realized in the core network, and partially on the radio side. Therefore, NFV MANO solutions are expected to orchestrate all these VNFs. Such an ever-increasing load on the NFV MANO solution may hinder the performance of MANO operations within the response time required to capture fluctuating KPIs. This phenomenon can be detrimental for V2X performance (e.g., increased response time from a V2X edge service to vehicle due to insufficient computing/network resources) and must be prevented. Further, the interplay between edge and cloud can be used by the NFV MANO solution to address the extra load incurred by network and vertical heterogeneity.

Insufficient and inconsistent input data Huge amounts of data are collected from surrounding infrastructure (edge computing nodes, sensors, vehicles) for orchestrators to coordinate distributed service deployments, which is more complex than in centralized clouds. This becomes more challenging due to the mobility and varying network connectivity, which may cause delays or jitters in data collection. This lack of sufficient and consistent input data leads to inefficiencies in decision-making, e.g., where/when to migrate service from one edge to another.

Support for multi-domain orchestration The access to V2X edge services should be ensured across different domains, as vehicles move along the roads, traversing from one edge domain to another. To this end, coordination among multiple orchestrators is required. Such MANO operations are performed across different NFV domains for particular V2X services

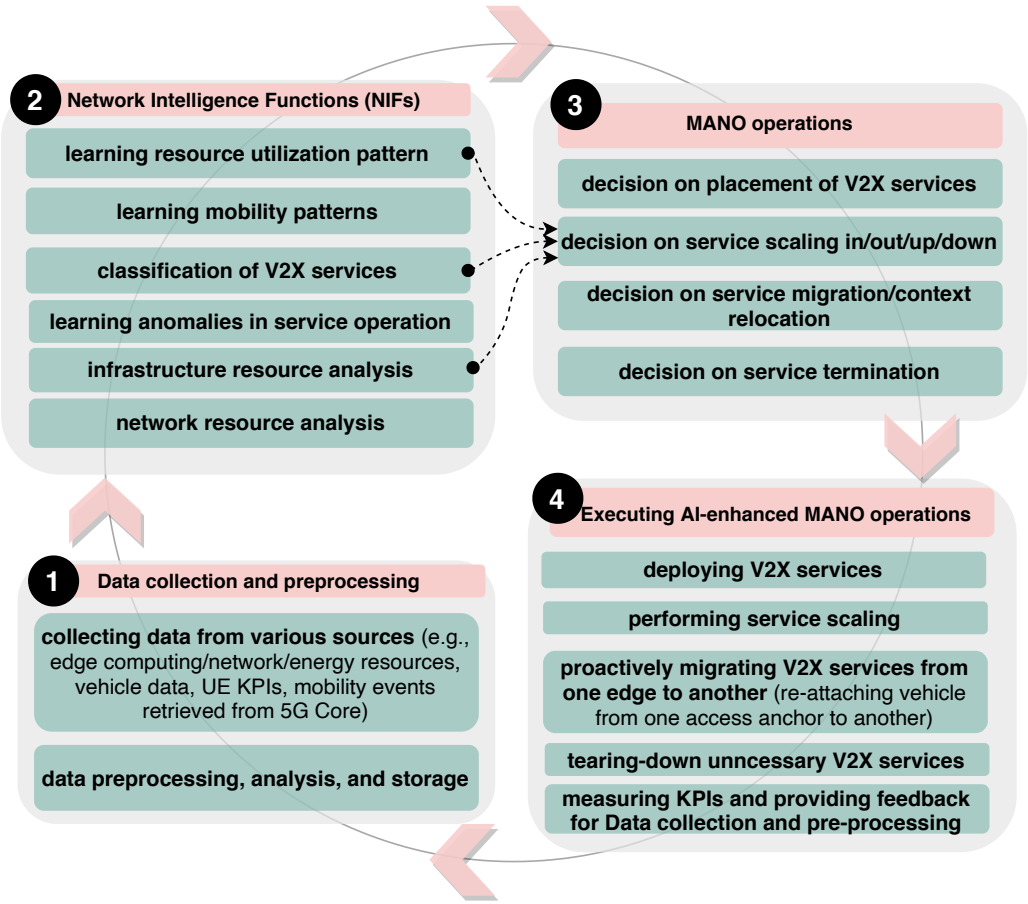


Figure 6.2: Closed-loop framework for NFV MANO in V2X systems; The dashed arrows showcase an example on how the decision on service scaling can be made based on the three different NIFs.

(e.g., services that send maneuver recommendations to vehicles in more than one domain to avoid road congestion), and can be realized by using particular learning framework.

6.1.2 The Need for Automated and Intelligent MANO for V2X

Addressing the previous gaps will transform traditional MANO for V2X systems into a fully autonomic system that is able to autonomously adapt the behavior of the services and infrastructure to respond to changes in user demands, business goals, and/or environmental conditions.

Unlike the legacy analytical-based models with too many configurable parameters that can affect KPIs, the data-based model introduces a data-oriented framework to realize the NFV MANO of V2X services, enabling a closed-loop approach to perform MANO operations (cf. Fig. 6.2), which is crucial for automation and optimization. This is precisely where AI/ML will play a fundamental role.

Moreover, this evolution toward automated MANO is in line with levels 3/4 of autonomous networks proposed by ETSI in [177], which require an automated distinction between different types of services, thereby analyzing the service performance and adjusting the service based on the changing conditions in the network. Furthermore, to mitigate the gaps listed in Section 6.1.1, we propose to integrate AI/ML techniques into a closed-loop framework to realize a fully autonomous NFV MANO, as these techniques are now sufficiently mature to provide efficient solutions, even for complex optimization and decision-making processes.

However, due to the complexity and heterogeneity of V2X systems, it is impractical to automate MANO operations by applying a single AI/ML technique or by creating a single ML model per MANO operation. On the contrary, suitable AI/ML techniques should be applied as NIFs, which focus on a particular task (e.g., mobility pattern, resource utilization), whose outcomes are then jointly considered in NFV MANO, where the final decision on how and which MANO operation to perform is made. Thus, in Fig. 6.2, we define the following phases of a closed-loop framework:

- *Data collection and pre-processing* is in charge of collecting data from various sources, which is then pre-processed and shared with the NIFs that apply corresponding AI/ML techniques.
- *NIFs* get the relevant data that is collected, and make predictions and decisions that support orchestrators towards improving their operations.
- *MANO operations* such as instantiation, scaling, migration, and termination, are performed based on the decisions that are considering and harmonizing outputs from a group of NIFs. For example, in Fig. 6.2, we showcase how the decision on service scaling should be made considering the outputs from learning resource utilization pattern, infrastructure resource analysis, and further adjusting the decision to a particular service class that is identified by the NIF that classifies V2X services.
- *Executing AI-enhanced MANO operations* is usually performed by MEC platform and virtualized infrastructure managers, which apply decisions made by orchestrators, and re-configure service deployments.

The closed-loop framework we propose is generic, but some widely used frameworks described in [98], such as Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K), and Observe-Orient-Decide-Act (OODA), can be applied. Notice that the actual mapping between the NIF and the different closed-loop framework blocks can vary depending on implementation.

6.1.3 AI/ML solutions for NFV MANO optimization and automation

To provide tangible ideas for the phases described in Section 6.1.2, we elaborate on the six examples for NIFs listed in Table 6.1. Table 6.1 also shows the data to be collected for the corresponding NIF, the potential AI/ML technique to implement it, and the list of the relevant V2X service types (cf. Fig. 6.1) that would be impacted by them. It is important to note that the orchestrator can also be realized as another NIF or using simpler approaches, e.g., ruled-based or multi-criteria decision making frameworks. The

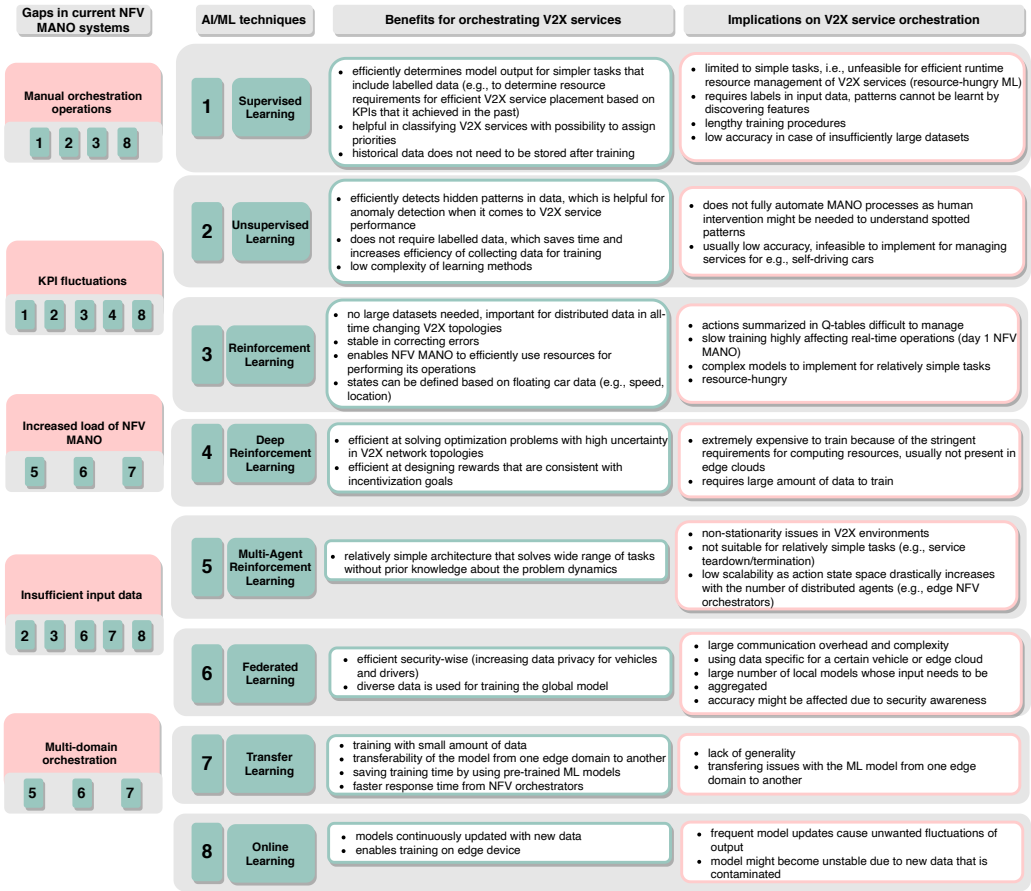


Figure 6.3: Overview of gaps in current NFV MANO solutions for V2X services, and potential solutions in the form of AI/ML models.

final decision by the orchestrator is then applied through exercising various resource reconfigurations (e.g., resource reservation/release).

Following, we will introduce each of the proposed NIFs (Phase 2 in Fig. 6.2), describing the reasoning behind them, and elaborate on the potential candidate AI/ML techniques to be applied, based on the analysis of eight well-known AI/ML techniques that are presented in Fig. 6.3 together with the gaps identified in Section 6.1.1 that these techniques are expected to alleviate.

6.1.3.1 Network Intelligence Functions (NIFs)

Learning resource utilization patterns for different types of service This task aims to learn the resource utilization patterns to i) determine the optimal resource requirements of a particular service type by looking for spatio-temporal correlations in historical data (Table 6.1), and ii) enable resource elasticity through forecasting resource utilization, which is important given the resource constraints at the edges. Based on the analysis shown in Fig.

6.3, Supervised Learning (SL) can use the labelled historical data to determine the relationship between edge computing and network resources that are provisioned to the service, and the KPIs measured at the client that consumes the service. Given this relationship, KPIs are determined based on the predicted resource utilization (e.g., regression models), and thresholds are defined to provide a finer-grained estimate of KPIs for a particular service when deployed at a certain edge node. Such a model can support the orchestrators in making decisions on service instantiation (e.g., edge node selection), or proactive scaling for maneuver recommendation services, and teleoperation services, as they are critical when it comes to service response time that needs to be monitored and granted.

Learning mobility patterns Learning the mobility pattern by applying SL/Uplink (UL) is beneficial for selecting the edge node to deploy the V2X service during the instantiation. This task needs to consider the data collected from the vehicles, thereby informing NIF about the speed, location, and heading of all connected vehicles, including data from the B5G core, e.g., mobility event notifications from Access and Mobility Management Function (AMF) [138], and looking for the spatio-temporal correlations of the vehicles' locations. In addition to using the mobility pattern to decide where to optimally place infotainment services, the decisions made in this NIF are provided as input for orchestrator to make a final decision on migrating V2X services. This is particularly important for collision avoidance services as they can be migrated to the edges closer to a dense group of vehicles that need to prevent collisions.

Classification of V2X services Service classification aims to select the network slices for each service, thereby improving the network QoS and QoE perceived by the vehicle clients and ensuring compliance with the Service Level Agreement (SLA). Therefore, SL can be used to classify on-demand V2X services (e.g., infotainment and teleoperation) and services that are always deployed on edge nodes (e.g., maneuver recommendation and collision avoidance). This task can also provide some sort of prioritization, so that these priorities can be considered when deciding which services to teardown/mute, and which ones to scale up to improve their performance.

Learning anomalies in service operation This task is expected to use Online Learning (OL) to identify anomalies during service operation. Take the infotainment service as an example, an offline-trained ML model may not respond effectively when there is a surge in the number of vehicles playing a specific live video stream. Thus, scaling operations are not triggered correctly, resulting in a decline in the perceived QoE. Another example is the decision-making model for scaling/migrating collision avoidance services, in which new data streams must be updated because vehicle collisions can occur sporadically. In the case of OL, models can learn in seconds and minutes, and update themselves based on new input data. This makes OL suitable for such NIF in V2X systems (e.g., data in motion), where new data streams from moving vehicles and network/computing infrastructure are constantly generated.

Table 6.1: Mapping the identified gaps to the proposed NIFs in the closed-loop framework for NFV MANO.

Data collection and pre-processing	Network Intelligence Functions (NIFs)			** MANO operations	V2X service type
	Role	Description	* ML		
edge computing resources, network resources, KPIs measured at client side	learning resource utilization patterns for different types of services	looking for spatio-temporal correlations in historical data, and forecasting resource utilization	SL, UL	I, M, S	infotainment, teleoperation, maneuver rec.
mobility events from 5G Core, floating car data (speed, location, heading)	learning mobility patterns	reducing dimensionality of multiple source information; finding spatio-temporal correlations of the vehicles' locations; scheduling V2X services to the neighboring edge nodes by precaching relevant content and balancing the load	SL, UL, FL	M	maneuver rec., collision avoidance
edge computing resources, network resources, KPIs measured at client side	classification of V2X services	mapping between QoS metrics and service priorities; assigning V2X service to a priority/non-priority slice	SL, UL	I, S, M, T	infotainment, teleoperation, maneuver rec., collision avoidance
KPIs measured at client side	learning anomalies in service operation	filtering anomalies as deviations from normal behavior; identifying the reckless driving maneuvers; isolating anomaly by allocating less resources to attacking sources	SL, UL, OL	S, M	infotainment, collision avoidance
edge computing resources, energy consumption	infrastructure resource analysis	evaluating computing resources /energy consumption trends for next operation hours; scheduling turning on/off the critical edge nodes according to computing resources/energy consumption plans	SL, FL, RL, MARL	I, M	infotainment, maneuver rec.
network resources (bandwidth, latency), mobility events from 5G Core	network resource analysis	predicting QoS metrics from current network state	SL, FL, RL, MARL	I, M	infotainment, maneuver rec.

* Supervised Learning (SL), Unsupervised Learning (UL), Federated Learning (FL), Online Learning (OL), Reinforcement Learning (RL), Multi-Agent Reinforcement Learning (MARL)

** Instantiation (I), Scaling (S), Migration (M), Termination (T)

Edge infrastructure and network resource analysis Inherently, this is a complex global optimization task given the ever-changing V2X topology and traffic fluctuations. Once thoroughly trained, Reinforcement Learning (RL) is robust and stable; thus, it is a promising technique for this NIF, especially because it uses an intelligent agent to learn by interacting with the environment in a closed-loop manner. Nevertheless, due to the high resource requirements, the application of RL should be carefully considered and used only to address the large-scale optimization of computing/network resources across multiple edge nodes. Thus, one solution is to use RL on the cloud-level orchestrators, where such a model can determine the state of the network traffic, e.g., determining the congestion zones for maneuver recommendation and infotainment services, and to help redirect vehicles, e.g., publishing recommendations/statistical analysis of congestion on the roads, in a common repository available to all edge orchestrators. Therefore, service placement/migration can be realized without supporting RL technique at each edge.

When collaboration between multiple edge orchestrators is required by V2X service, e.g.,

cooperative maneuvering of (automated) vehicles, Multi-Agent Reinforcement Learning (MARL) can deploy multiple learning agents at different edges that interact to solve a problem. An important benefit of MARL is its relatively simple architecture that solves a wide range of tasks without prior knowledge of the problem dynamics. This is important for continuously changing environments with computing and network resource fluctuations. However, if the cloud orchestrator deploys a V2X service on a particular edge node that lacks collected data from edge and network infrastructure, Federated Learning (FL) can apply a global ML model, which has been thoroughly evaluated in local edge domains based on their data, to help predict resource usage on this particular edge node. Moreover, FL can bring privacy to collect data in distributed edges.

6.1.3.2 Implications of applying AI/ML in MANO for C-V2X systems

Besides introducing automated and intelligent MANO for V2X systems, AI/ML techniques impose additional challenges that shall be carefully considered.

Quality of data The performance of AI/ML techniques in decision-making (e.g., prediction, classification, MANO operations) depends on how close the training data is to the actual data used in the production environment. The lack of real-world samples may impose an unmeasurable risk when training ML models based on synthetic data due to the risks to driver safety. However, collecting real-world data is time-consuming, as scenarios that require specific MANO operations (e.g., service scaling during natural disasters) are difficult to replicate and repeat multiple times to collect sufficient data for (re)training.

Security, scalability, and transferability These are potential factors limiting AI/ML techniques for NFV MANO in B5G V2X system [174, 178]. In general, AI/ML solutions are only as reliable as the data upon which they are trained. This is especially important as some V2X services need to assist their users through potentially life threatening situations. In terms of security, particularly for vehicle data (e.g., vehicle identification, location, destination, and speed), one possible solution is to apply an advanced identity and access management framework where vehicles are authenticated, authorized, and audited, and are represented by security tokens that are stored only on specific edge servers. Regarding scalability and non-stationarity issues in RL and MARL, the former is due to the drastic increase in the action state space as the number of agents increases, and the latter is due to the decisions being influenced by the actions of other agents. An example is the relocation of emergency services based on the action taken by the source edge orchestrator, whereas the target edge orchestrator decides to mute all other services completely due to existing prioritized maneuvering operation. This situation must be carefully monitored and prevented, to avoid conflicting decisions made by different orchestrators.

High computational power Resource-constrained edge nodes may not be able to offer high computational power, which makes them unsuitable for running heavy data-processing tools (e.g., Apache Spark and deep learning libraries), but rather for lightweight ones. Nevertheless, the imbalance between lightweight implementation and high performance requires

further study. Given the edge resource constraints, the applicability of most ML models is limited. Some efforts have been made to reduce computational and memory loads by applying network compression, i.e., pruning mechanisms [176]. An example is provided in [179], in which a Tiny ML (TML) based on incremental learning is built by performing training and inference directly on the device. In contrast, although there are some approaches for training models in the cloud, they are error-prone because these trained models often do not correctly reflect the edge environments.

Proactive fault tolerance Despite the prediction capabilities of intelligent MANO, their accuracy remains challenging. Since every pattern has exceptions, like outliers in the data, the proactive MANO operations may make incorrect decisions when such prediction errors occur. Thus, it is necessary to study the extent to which predictors may make mistakes and determine whether they have serious consequences for service performance and whether plan B (e.g., reactive approach) should be prepared.

This might not be a significant challenge for infotainment and maneuver recommendation services, as their demands can be tested from a large number of vehicles, even with small computing units that can access the network and retrieve data (e.g., video content or route notifications). However, teleoperation and collision avoidance services require careful planning and preparation for data collection and testing. One possible approach to address this challenge is to use digital twins that mimic the real environment, so that algorithms can be trained in a safe environment, but close enough to the environment where they are deployed.

NI Orchestration Layer NIFs will be empowered by AI/ML algorithms, which will require a different life-cycle management compared to edge V2X services or VNFs in general (e.g., model training, loss function adaptation, and resource-awareness). To fully support a complex, pervasive, and distributed nature of NI, a NI Orchestration layer should be introduced to manage intelligence as a whole, ensuring the ideal functioning of each closed-loop NIF, and overseeing interactions across closed loops that run NI at different timescales[175].

Update frequency of ML models Although OL can be applied to learn anomalies during service operation, too frequent model updates may cause unwanted fluctuations in model output. However, if the update frequency is too low, catastrophic forgetting can occur, where previously learned knowledge is forgotten due to non-stationary data. Therefore, we expect the NI orchestrator to continuously monitor the performance of such models and the quality of their decisions, thereby adjusting the frequency of model updates based on the vehicle location and mobility pattern.

Modular design of B5G C-V2X applications It is important to properly modularize the overall V2X service in a set of loosely coupled applications that can be migrated/scaled according to the decisions made by MANO orchestrators. To support dynamic V2X environments, applications should rely on middlewares providing location-transparent communi-

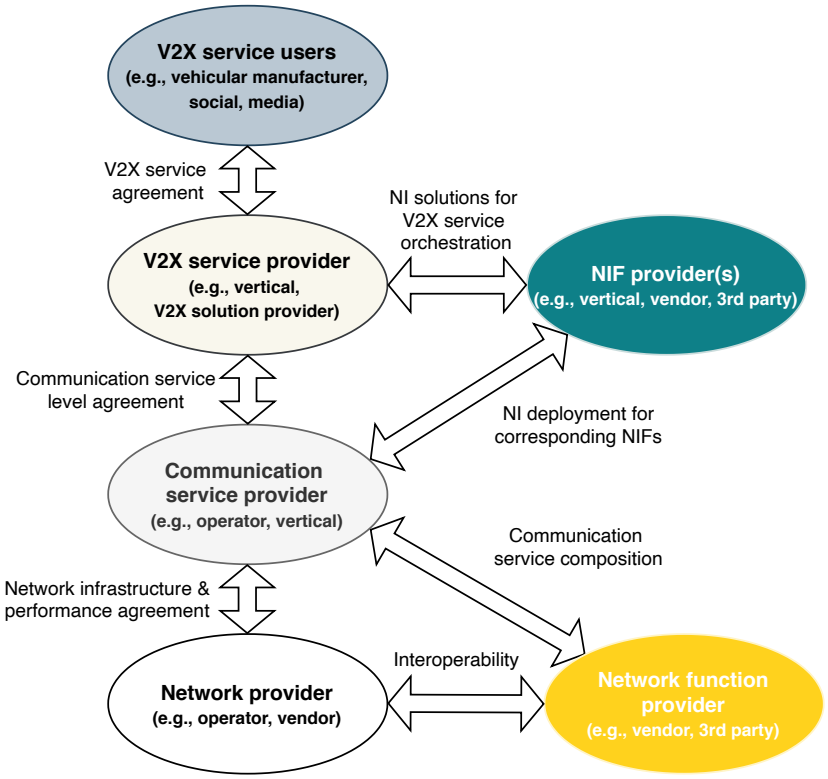


Figure 6.4: Different providers in the value chain for V2X industry.

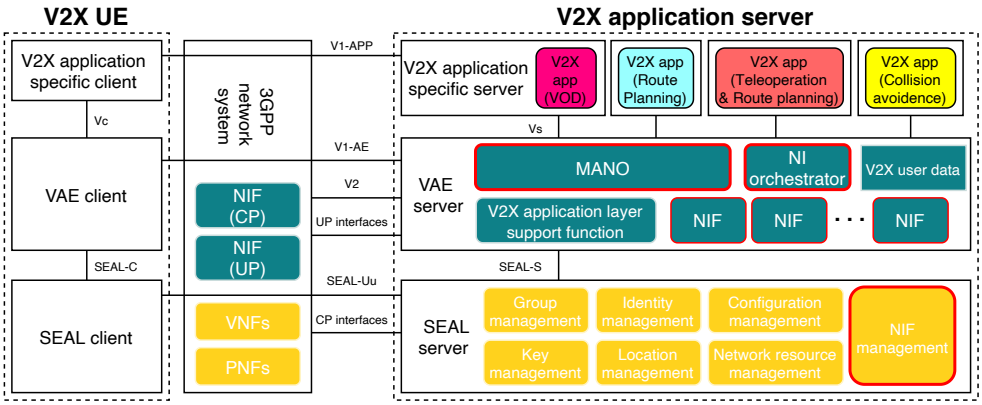
cation and data access (e.g., Zenoh[1]) that is not hindered by the ever-changing underlying network topology and infrastructure. Also, V2X services dealing with real-time constraints should be built on top of time-aware framework (e.g., zenoh-flow [180]) to react effectively to any event in the system (e.g., network error, server congestion). This allows critical applications and MANO to fall back to default safe mode, e.g., when enhanced V2X at the edge suffers from unpredictable performance, autonomous vehicles may slow down while the MANO migrates/scales the involved V2X applications.

6.1.4 Network Intelligence in V2X ecosystem

In addition to the above introduced NIFs for MANO in V2X systems, we further examine the essential elements for implementing an overall NI system.

To introduce how NIFs fit the V2X ecosystem, Fig. 6.4 illustrates the relationship between different providers in the value chain transformation of the V2X industry, in line with 3GPP TS23.286 [181]. First, providers of network infrastructure, network functions, communication services, and V2X services, are decoupled from V2X service users to allow cost-effective and flexible V2X service composition. Additionally, a new role is expected to provide NIFs in the form of AI/ML algorithms outlined in Section 6.1.3. Specifically, these NIFs should not

[1]Zenoh framework: `https://zenoh.io/`



Functional entity & Reference point	Descriptions
V1-APP & V1-AE	V1-APP supports interaction related to V2X applications, V1-AE supports interaction related to V2X application layer support functions.
V2	Between VAE Server and V2X control function. The V2X control function provides UE parameter for V2X communication.
Vs, Vc	Support interactions related to V2X application layer support function and V2X application specific server/client.
SEAL-S, SEAL-C, SEAL-Uu	Include a number of reference points for respective management in SEAL architecture.
V2X application specific server/client	Provide server/client side functionalities (e.g., platooning server) corresponding to the V2X applications.
VAE server/client	V2X Application Enabler (VAE) server/client provides the client/server side V2X application layer support functions.
SEAL server/client	Various Service Enabler Architecture Layer (SEAL) services offer certain functionalities as APIs for vertical applications.

Figure 6.5: V2X application layer functional model with NIFs, NFV MANO, and NI orchestrator.

only be supported at MANO layer but also for the corresponding network slice(s) in Control Plane (CP) and User Plane (UP). Take the second gap in Section 6.1.1 as an example, deploying the NIFs for MANO operations (e.g., scaling) may not be sufficient to respond promptly to KPI fluctuations, and V2X service users can notice performance degradation (e.g., 3GPP alternative QoS profiles). To this end, NIFs in CP and UP can play a key role in adjusting scheduler policies and manipulating packets (e.g., packet marking/dropping), respectively.

Moreover, we propose an enhancement for the V2X application layer functional model from 3GPP, to be able to manage NIFs for V2X services and corresponding network slices, as shown in Fig. 6.5. We can see that the V2X application layer support functions at the V2X Application Enabler (VAE) layer exploit several SEAL services to support V2X applications operations (see 3GPP TS23.286 and TS29.486). We propose two additional functional entities: i) NIF management service at the SEAL server, and ii) NI orchestrator at the VAE server. The former service can be exploited by means of SEAL server APIs to interact with 3GPP network system for modifying NIFs. The NI orchestrator can provide support functions to communicate the requested NIFs to the 3GPP network and manage the applied NIFs accordingly to fulfill implications mentioned in Section 6.1.3.2. Finally, this NI orchestrator can harmonize different NIFs mentioned in Table 6.1 (e.g., V2X service classification, anomaly detection) and realize a fully data-driven MANO for V2X services.

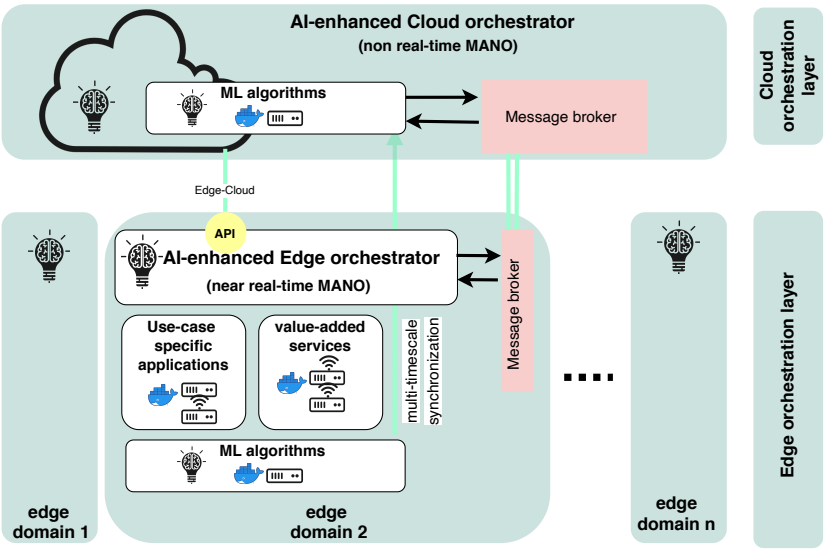


Figure 6.6: The architecture of multi-domain AI-enhanced management and orchestration system for V2X use cases.

6.2 Leveraging AI/ML techniques to automate and enhance MANO systems

As elaborated in Section 6.1, there is still a gap in research when it comes to experimentation and testing the true impact of AI/ML on the optimization of NFV MANO operations, as state-of-the-art work is either based on other optimization techniques (Lyapunov optimization techniques [182]) that might be complex and lengthy for service management in V2X systems, or their evaluation is based on the simulations [183, 184].

To this end, in this Chapter we present our work towards building and fully utilizing the potential of high-performance real-life testbeds, such as Smart Highway[2] [185] and Virtual Wall[3], to pursue testing and validation of distributed intelligence in a dynamic network such as V2X system. We present the AI-enhanced MANO system for V2X services in Fig. 6.6, with cloud and edge orchestration layers, which are enabled to autonomously operate, but also to collaborate and balance their operations towards achieving desired KPIs.

6.2.1 Realistic Experimentation Environment for AI-enhanced MANO of 5G and beyond V2X systems

The AI-enhanced MANO system for V2X services that we present in this section, and illustrate in Fig. 6.6, consists of two layers, i.e., cloud and edge. The system enables autonomous MANO operations in each of the domains, but enforces an interplay between

[2]Smart Highway: `https://www.fed4fire.eu/testbeds/smart-highway/`
[3]Virtual Wall: `https://www.fed4fire.eu/testbeds/virtual-wall/`

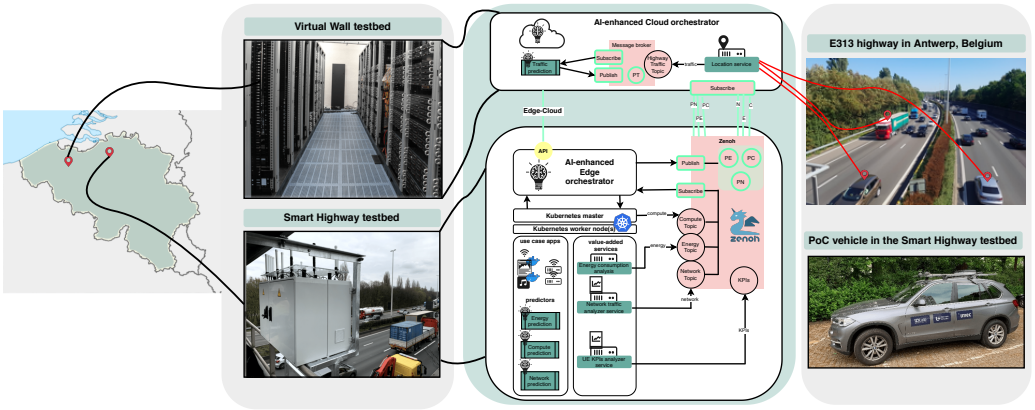


Figure 6.7: The AI-enhanced management and orchestration system mapped to the real-life testbed environment (PE - pre-processed/predicted energy, etc.).

them for offloading orchestration decisions, or for retrieving data from distributed data engineering pipelines available in all edge domains. Such a system orchestrates both services and applications developed for various use cases, but also NIFs that are represented by adopted and integrated AI/ML models.

Despite the emerging popularity of bringing intelligence to network management and orchestration functions in 5G and beyond, most of the works on validating the impact of AI/ML on MANO are based on simulations. There is a gap between using synthetic data and real data when it comes to training and validating/testing AI/ML models, as real setups can create more realistic traces for training, with higher probability of good performance when deployed in production environments. However, building realistic PoCs is usually time-consuming and expensive, while the number of scenarios that can be covered is limited. On the other hand, simulators bring that flexibility but mostly at the cost of not capturing all dynamics of real environments. Thus, the real setups are fundamental to create hybrid approaches that ensure that the performance of AI/ML algorithms is not negatively impacted once they are dealing with real data. One of the attempts to pursue testing of AI/ML on the lifecycle management operation of scaling service functions is presented by Baranda et al. [186], where a scaling operation of vCDN service is triggered by AI/ML algorithms, thereby integrating AI/ML into management platform of 5G-Transformer[4]. Thus, in this section, we present and illustrate a realistic experimentation environment that extends the scope of aforementioned PoC, and enables studying and experimenting with AI-enhanced operations of proactive placement, scaling, migration, and termination, of challenging V2X services, towards understanding and resolving challenges imposed by AI/ML to overcome them and improve those MANO operations.

[4]5G-Transformer - the project on 5G Mobile Transport Platform for Verticals: `http://5g-transformer.eu/`

6.2.1.1 Architecture of AI-enhanced management and orchestration system

The architecture of multi-domain MANO system presented in Fig. 6.6 is applicable to all distributed and heterogeneous softwarized networks whose operation stretches from edge to the cloud, where services and EdgeApps are usually deployed with microservice-based approach, and connectivity ensured via different wireless technologies including 5G and beyond. As such networks are usually characterized by distributed resources belonging to different edge domains, which might belong to different MNOs, we follow the split between cloud (i.e., centralized) and edge orchestrators, which are deployed in a relationship $m : n$, $m < n$, $m, n \in \mathbf{N}$.

Thus, each edge domain that consists of one or multiple edge nodes (i.e., MEC hosts) is governed by one edge orchestrator, which is, following ETSI NFV MANO framework, in charge of lifecycle management (e.g., instantiation, scaling, and termination) of all underlying services, i.e., i) use case-related services, ii) value-added services, and iii) NIFs that embody AI/ML models. On the other hand, cloud orchestrator is rather in charge of global optimization in the system, thereby making less-granular decisions depending on the e.g., locations and density of vehicles on the roads for our particular real-life use case. One particular example of these decisions is service migration from one edge domain to another (described and exemplified in Section 6.2.2, triggered by higher density of vehicles (i.e., edge service consumers) in one edge domain, or by need for optimization of energy consumption in MEC hosts across edge domains.

Two MANO layers communicate with each other in the two following ways: i) via Edge-Cloud reference point, which is used to either offload decision-making tasks between two orchestrators or to pass the already taken decision, and ii) via message brokers, which exchange data in a controlled way depending on the type of AI/ML technique that has been applied in the system, thereby using that data to either perform training or model adjustments and online learning. Thus, depending on the time-scales of optimization (global or local, i.e., edge-specific), it is required that MEC hosts can connect data to AI/ML models in a transparent and efficient way (e.g., using Zenoh framework introduced in Section 6.2.1.2). In case of federated learning, which is suitable for distributing intelligence across edge nodes, thus deploying AI/ML agents in edge nodes, we consider that each edge orchestrator trains the local model based on the data collected from its own domain. On the other hand, if security in data sharing between two message brokers laying in two orchestration layers can be preserved, multi-agent reinforcement learning may use data collected from other edge domains to optimize policies.

6.2.1.2 Proof-of-Concept

In Fig. 6.7, we map the testbed components to the elements of AI-enhanced MANO framework presented in Section 6.2.1.1. Starting from the edge, we provide the NFV infrastructure in MEC hosts by virtualizing computational resources in RSUs, which are deployed along the E313 highway in Antwerp, Belgium, as a part of the Smart Highway testbed [185]. The MEC nodes are collocated with RSU units, as presented in our paper [187], and used it in the demo setup for emergency V2X services in [188]. To make use of the computational resources for performing lifecycle management of edge V2X services, we deploy K8s, where

edge orchestrator embodies the role of K8s master and extends it to i) support cross domain operations, i.e., edge-cloud interaction, and ii) receive dynamic triggers from AI/ML models deployed in NIFs for optimizing MANO operations. Such K8s master with extended and enhanced operation deploys services and applications on designated worker nodes. In the PoC, both master and worker nodes can be deployed on the bare metal, as well as in LXC, which is a more suitable practice for shared experimentation environments as testbeds.

For each type of data that is collected, i.e., computational and network resource utilization, energy consumption, KPIs measured at users' side, and users' locations, we also deploy MEC value-added services, as per definition in ETSI MEC [50], which perform data retrieval and pre-processing before publishing them on Zenoh [189]. Given its minimal network overhead (as little as 5B), and its small footprint (around 60kB on Arduino board), Zenoh is adopted in our PoC as a framework for data engineering pipeline. In particular, Zenoh provides a minimal set of primitives to deal with data in motion (e.g., real-time stream of vehicles' location/speed/destination), data at rest (e.g., historic data for vehicles' and edge nodes' computational resource utilization and energy consumption) and remote computations (e.g., on-demand calculation of the best route and speed limit). Each edge and cloud orchestrator acts as a subscriber for various types of data that can be stored on edges, and used for training or online learning/optimization.

Furthermore, concerning the vehicle as a client, our current PoC includes one vehicle that is capable to communicate with the edge services via long range 4G (to be extended to 5G in future). Thus, the client application is installed in the OBU of the vehicle, and it utilizes Uu link to exchange Cooperative Intelligent Transport System (C-ITS) messages with services, and inform them about its location, speed, heading, and destination.

Cloud orchestrator is running on the bare metal on top of the Virtual Wall testbed, located in Ghent, Belgium (Fig. 6.7). It is deployed as a web server (using Flask framework in python), which is capable of i) processing decision-offloading requests coming from the edge orchestrators, ii) location data processing and publishing on Zenoh, iii) injecting decisions on the north-bound interface of edge orchestrators to instruct them to proactively migrate/relocate services from one edge to another, and iv) receiving notifications from NIFs deployed on the cloud, which enhance their operations and help them make efficient decisions on managing underlying resources and edge orchestrators.

In Fig. 6.8 we show the result of average response time, and CPU utilization, of the vCDN server deployed on the MEC host in our PoC. To stress the load and increase the number of vehicles, we run Locust[5] stress test inside the vehicle. We can see that the number of vehicles that are simultaneously requesting content from the same server affects the response time, and CPU utilization as well. In case NIF predicts the traffic demand, and the number of vehicles in this specific geographic region, they are expected to optimize the operation of an edge orchestrator, as it will perform horizontal scaling and additional deployments of vCDN server on other MEC hosts, so that users (i.e., vehicles) can still experience low response time. As the response time consists of communication latency (uplink and downlink, impacted by network load), and computational latency (affected by CPU load), its increase is mainly affected by an increase in CPU utilization on edge nodes, which needs to be carefully monitored and optimized e.g., by corresponding NIFs. In Section 6.2.3, we further study the

[5]Locust: `https://docs.locust.io/`

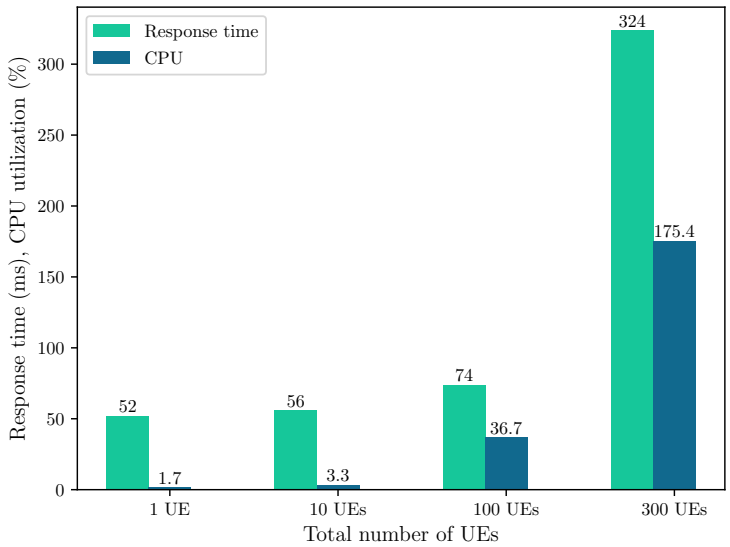


Figure 6.8: Average response time and CPU utilization of vCDN server deployed in our PoC.

relationship between the KPIs measured at the client side, and the infrastructure metrics such as CPU utilization, and present how this relationship can be leveraged to proactively perform orchestration operations that will improve the QoS experienced by the end user.

We presented the realistic PoC in IEEE Consumer Communications & Networking Conference, and in the demo[6], we have shown i) the enhanced capabilities of the testbed to monitor and collect data, ii) the enhanced interfaces towards the orchestrators to consume and preprocess the data, and iii) an intelligent algorithm performing a MANO task to change the behavior of the system in an autonomous (closed-loop) way.

6.2.2 An optimized application-context relocation approach for Connected and Automated Mobility (CAM)

In 5G-based vehicular systems, a vehicle is capable to collect the contextual driving information, thereby connecting to the vehicular services and EdgeApps, located at the edge in order to keep the communication latency to a minimum possible level. In particular, to be less dependent on driver's actions, and to ensure higher safety, the vehicle needs to receive instructions from the network infrastructure in less than 100ms [190], which requires service availability close to the vehicles, i.e., in the edge infrastructure such as MEC platforms, as well as transferring the application traffic via 5G Uu interface [191]. Thus, in this section, we present a management and orchestration framework that enables service continuity in a highly mobile environment, with the reference to the 3GPP architecture for enabling edge applications [192], and ETSI NFV MANO framework [140]. The service continuity is enabled via an optimized application-context relocation approach that is triggered by a MEC

[6]The demo video can be viewed on the following link: `https://drive.google.com/file/d/1EFn5Lwwrvsre1hTiMllpizkvrtD3hfYn/view?usp=sharing`

application orchestrator while a vehicle, which is a consumer of the CAM service on the edge, moves along the road.

To efficiently solve the challenges on how and when to perform application-context relocation, the MEC orchestrator in our framework is performing the prediction of resource availability in edge NFVI, utilizing the prediction model based on Long Short-Term Memory (LSTM) [193], and making a decision on the optimal application service placement by running the The Technique for Order of Preference (TOPSIS) algorithm, i.e., one of the widely adopted Multi-Criteria Decision Making (MCDM) concepts [194], thereby taking into account: i) the aforementioned resource availability prediction, ii) the latency and bandwidth on the communication path to the vehicle, and iii) geographical locations of vehicle and MEC host in the edge infrastructure. To measure the performance of the MEC application orchestrator, we have leveraged a PoC of the management and orchestration framework in a real-life distributed testbed environment, which is described previously in Section 6.2.1.2, with a slight variation on the selected nodes that are utilized for service deployments (more details shown in Fig. 6.12).

Since the autonomous vehicles need to continuously collect the data from surrounding environment and network infrastructure, including the suggestions on braking and accelerating without driver assistance, the experimentation in our PoC reflects such a use case in which MEC application service is informing vehicle about driving conditions on the road (e.g., traffic jams, poor weather conditions, emergency situations, etc.). Thanks to the distributed service deployment, vehicle is being informed about driving conditions not only in its close proximity, but also in extended regions, thereby enabling vehicle to choose another route for its maneuver. In this section, we show the improvement of the response time when application-context relocation is performed, thereby proving the efficiency of the MEC application orchestrator in optimizing the MEC host selection and application-context relocation towards achieving service continuity.

6.2.2.1 Edge-aware Management and Orchestration framework

As a part of Release 17, 3GPP is standardizing an architecture for enabling edge applications, while providing mutual awareness between edge client applications (i.e., in-vehicle application), and edge application servers running in the edge data network. This 3GPP standardization track [192] created i) the application layer architecture, which is shown in Fig. 6.10, ii) procedures, and iii) information flows necessary for enabling edge applications over 3GPP networks. In particular, in architecture shown in Fig. 6.10, the edge network consists of i) Edge Configuration Server (ECS), which provides configuration data, i.e., Local Area Data Network (LADN) URI, to the Edge Enabler Client (EEC) to connect to the Edge Enabler Server (EES), ii) EES, which interacts with 3GPP core to collect network and service capabilities (e.g., location services, QoS management, etc.) that will improve the performance of edge application server, thereby enabling Edge Application Client (EAC) to connect to the server, and iii) Edge Application Server (EAS), which performs server functions and exchanges application data traffic with the client (Figures 6.9 and 6.10). On the client side, in our case in the vehicle, EEC discovers the edge network, retrieves the necessary information for connecting to the edge (e.g., coverage area/service area, types of application servers or MEC applications, etc.), and connects to it via IP address provided by

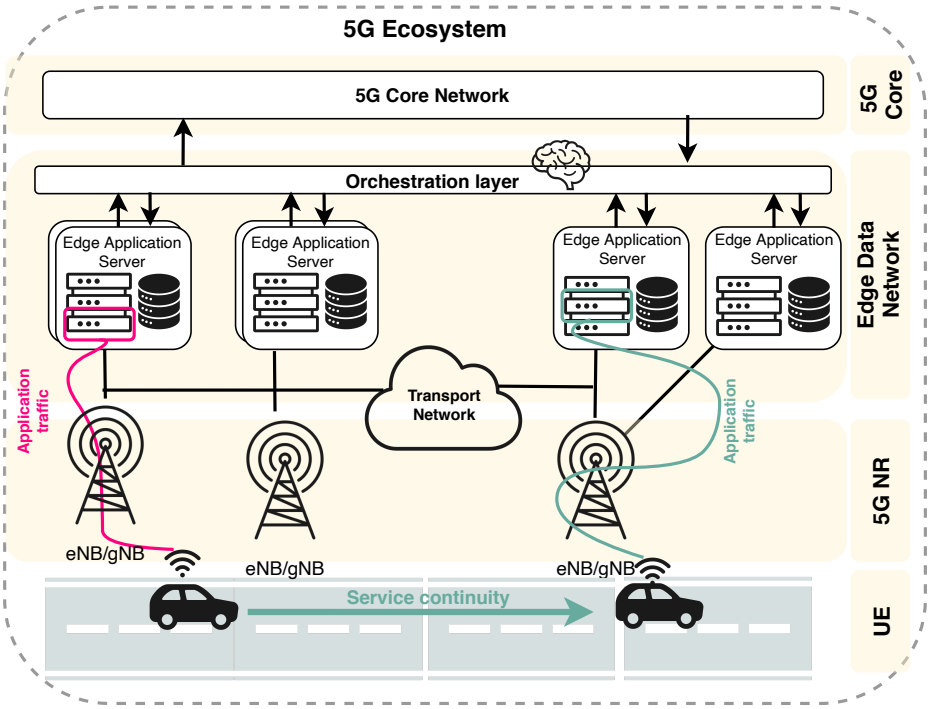


Figure 6.9: Enabling service continuity for vehicles in 5G ecosystem.

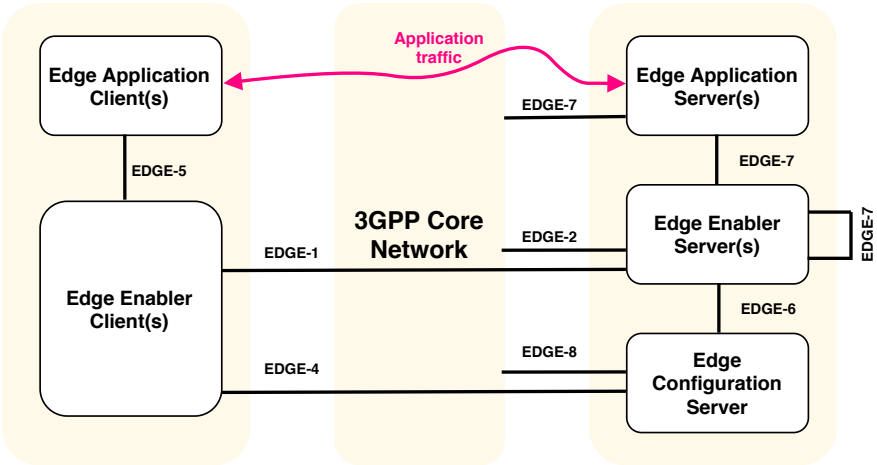


Figure 6.10: 3GPP Architecture for Enabling Edge Applications.

EES. Furthermore, different reference points, i.e., EDGE 1-EDGE 7, are defined to enable communication between different architecture elements.

In Fig. 6.11, we present the message sequence chart to showcase the operation of the application-context relocation from one edge to another, thereby mapping our management and orchestration framework (black boxes on the top), which is based on ETSI NFV MANO [140] and presented in [187] and Section 6.2.1.1, to the 3GPP architecture for enabling

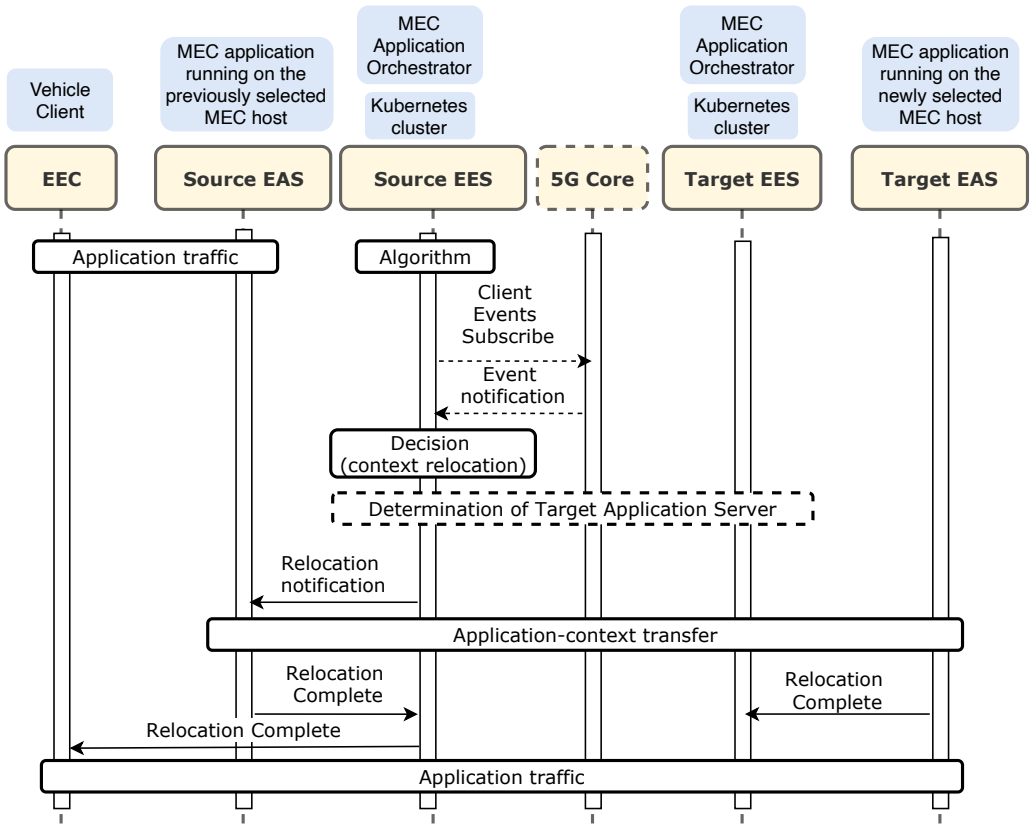


Figure 6.11: Message Sequence Chart for the application-context relocation procedure.

edge applications (yellow boxes). In particular, when vehicle sends a discovery request to the edge or MEC orchestrator, as a response, it receives a list of all available MEC application services that corresponds to the filters applied in the request. This way, the vehicle becomes edge-aware, as it can connect to any application server from the list. Once MEC orchestrator decides that vehicle needs to connect to another MEC application service due to e.g., increased resource consumption that will degrade the QoS, vehicle going out of the geographical service area, vehicle re-attaching from one UPF anchor to another, etc., the same reference point, i.e., EDGE-1, is used to inform vehicle about the newly selected MEC host (i.e., Relocation complete notification in Fig. 6.11). Furthermore, this notification contains the endpoint of the new MEC application instance running on the new MEC host, and client in the vehicle needs to be configured in the way that it can dynamically change the IP endpoint of the application server from which it consumes the service.

6.2.2.2 Optimized MEC host selection

To transfer the context of application service that vehicle is consuming, and to enable this vehicle to continue utilizing the service in a seamless way, we need to i) identify a corresponding target MEC host, ii) perform transfer of application-context, iii) reconfigure the

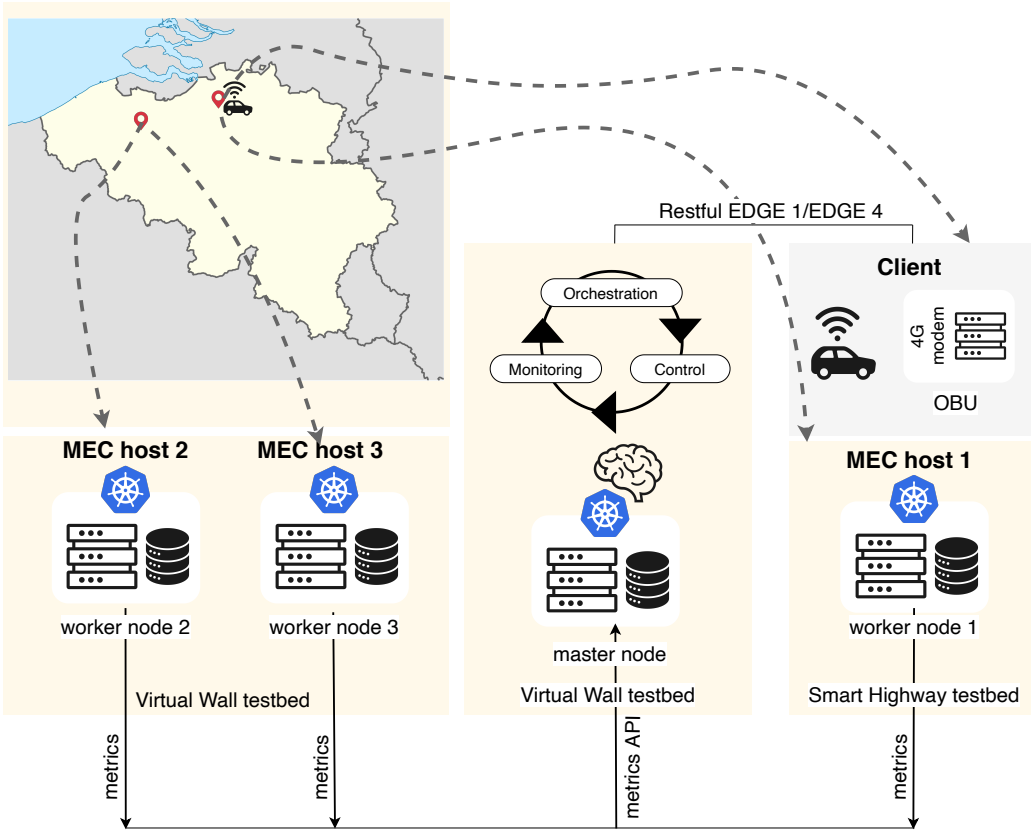
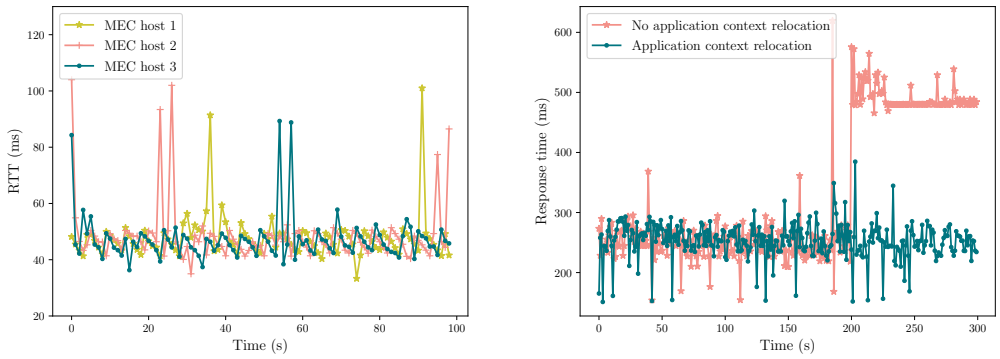


Figure 6.12: A more detailed view on the PoC utilized in Section 6.2.2.

traffic rules and management policies, and iv) setup a new communication path to the vehicle. The step i) is performed by our MEC application orchestrator that is designed as an extension of Kubernetes master role. It runs the optimized MEC host selection algorithm, thereby predicting the resource availability in all MEC hosts that belong to the management and orchestration framework, by applying the LSTM based prediction. Furthermore, taking into account the predicted resource availability, the latency and bandwidth on the communication path to the vehicle, and geographical location of both vehicle and MEC hosts, the orchestrator makes decision whether application-context needs to be transferred to another edge or not, by performing the MCDM analysis. If the decision is made, and new node is selected for application placement, orchestrator instantiates new application service on the target MEC host, and allows application services from the source host to transfer the context to the target host, as shown in Fig. 6.11. Finally, once the context is transferred, the orchestrator sends a notification to the edge-aware client application in the vehicle, which then starts consuming service from the new MEC host, after the traffic rules and management policies are reconfigured by the MEC application orchestrator.



(a) Round-trip time. *(b) Overall response time.*

Figure 6.13: Optimized MEC host selection results.

Table 6.2: The mean and standard deviation values for two scenarios.

Scenario		Mean (ms)	Standard deviation (ms)
1	No application context relocation	331.117	117.543
2	Application context relocation	252.924	29.786

6.2.2.3 Results

The application service running on the distributed MEC hosts in our PoC are cloud-native Docker-based applications deployed in Kubernetes environment, with RESTful APIs exposed to vehicles for retrieving information about driving conditions on the road in a JavaScript Object Notation (JSON) format.

In Fig. 6.13a, we show the trace of the measured Round Trip Time (RTT) values for the client running in the vehicle on the Smart Highway, and for all three application servers deployed in distributed MEC environments, in order to test the impact of the network on the overall service response time, which contains the transmission and propagation delay (network impact), and computational delay on the application server (MEC impact). Furthermore, in Fig. 6.13b, we show the overall response time of the application server, measured on the client side, for two different scenarios. This response time is important because it shows the delay in retrieving the important contextual driving information from the server, and keeping this response time at a low level (e.g., below 100 ms) is essential for vehicle to make decisions.

In both scenarios, the MEC host 1 is never selected by MEC application orchestrator for an application placement due to the high resource consumption (since we have increased it artificially by performing load stress tests to train our prediction model), while MEC hosts 2 and 3 are being selected based on the projected resource consumption due to the RTT of similar scale. In the first scenario no application-context relocation is performed, thus, vehicle remains connected to the MEC host 2, and as it can be seen in Fig. 6.13b, once the load increases on the MEC host 2 (after 200 s), the response time of the application service

is increasing, which means that the driving information about the conditions on the road might be significantly delayed at the vehicle side, leading to the inefficient decisions that will affect the whole maneuver experience. On the other hand, in scenario 2, we show that in the case when load increases on the MEC host 2 (i.e., resource availability decreased), as predicted by our algorithm for the time after 200s, the proactive decision on relocating the application-context from application service on the MEC host 2, to MEC host 3, results in the relatively stable response time, which does not increase when vehicle starts retrieving service information from application service on the MEC host 2. Furthermore, a similar decision can be made by our algorithm in case user mobility event notification is received from the core network, and testing such scenario is part of our future work.

The mean and standard deviation values for both scenarios are shown in Table 6.2, and we can see that in scenario 1, when there is no application context relocation for the observations that appear after 200th second, the deviation from the mean is large, i.e., the increase in response time is statistically significant. Thus, in scenario 2, we show that optimized and proactive MEC host selection that results in application-context relocation helps to improve the overall response time, and to prevent service unavailability that leads to outdated information about the conditions on the road, which consequently highly affects the maneuver decisions made by vehicle.

6.2.3 MAESTRO algorithm

In this section, we present our ML-based quality-aware concept that automates edge service orchestration, and we utilize the high-performance real-life testbeds, such as Smart Highway[7] [185] and Virtual Wall[8], to pursue testing and validation of such concept in a dynamic network such as V2X system. To measure the performance of the created algorithm, we have leveraged the PoC of the management and orchestration framework described previously in Section 6.2.1.2.

In Fig. 6.14, we illustrate the deployment of ML-driven MANO system for V2X services, with cloud and edge orchestration layers, which are enabled to autonomously operate, but also to collaborate and balance their operations towards achieving the desired KPIs. The Ml-enhAnced Edge Service oRchestration (MAESTRO) algorithm we created is a hybrid edge service relocation algorithm, which is a MCDM algorithm based on TOPSIS [195] and Support Vector Regression (SVR) [196]. This model is trained at the edge orchestration layer, and it uses collected data to learn the interrelation between the infrastructure and service performance metrics, and to predict the average response time of a V2X service running on the edge computing node (e.g., RSU in our PoC deployment illustrated in Fig. 6.14). Further, the model is used by the cloud orchestrator to proactively decide on whether the service relocation should be performed from one edge to another, in order to avoid service disruptions due to mobility and low service performance.

With the performance analysis we conducted using this realistic testbed setup, the contribution of this work is two-fold: i) we study the interrelation between MEC infrastructure (measured at NFVI) and service performance metrics that are being monitored by the cloud

[7]Smart Highway: `https://www.fed4fire.eu/testbeds/smart-highway/`
[8]Virtual Wall: `https://www.fed4fire.eu/testbeds/virtual-wall/`

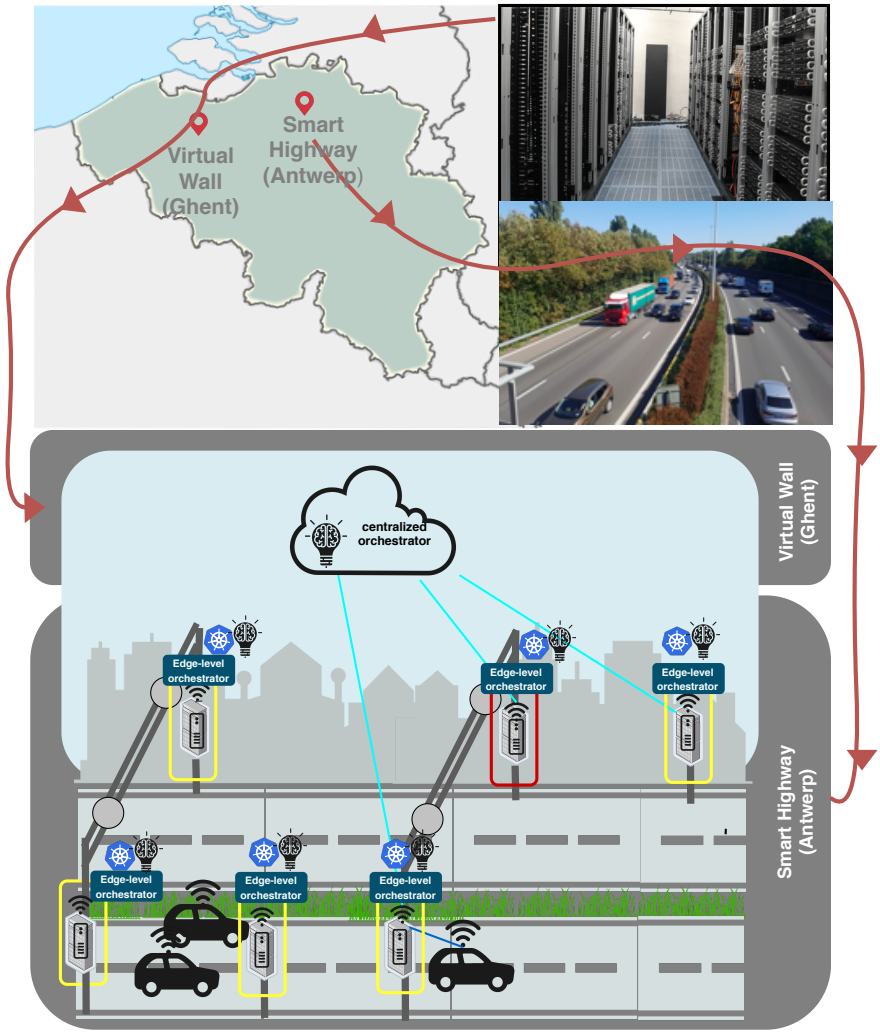


Figure 6.14: The multi-domain AI-enhanced management and orchestration system for V2X use cases.

and edge orchestrators (measured at client side), and ii) we propose and evaluate an ML-based quality aware algorithm, i.e., MAESTRO, to automate edge service orchestration, thereby minimizing average service response time, while ensuring high service availability and reliability. As described in Section 6.2.1.1, our NFV MANO framework consists of two layers, i.e., cloud and edge, which perform autonomous MANO operations, but enforce an interplay between them for managing orchestration decisions, or for retrieving data from distributed data engineering pipelines available in all edge domains. To enable the cooperation between different orchestration entities, certain management-level agreements need to be ensured, and more information about this type of agreement can be found in our previous work [169], i.e., in Chapter 4.

This intelligent NFV MANO framework is suitable for orchestrating edge deployments of V2X services and EdgeApps that require low-latency and high-reliability (e.g., service continuity

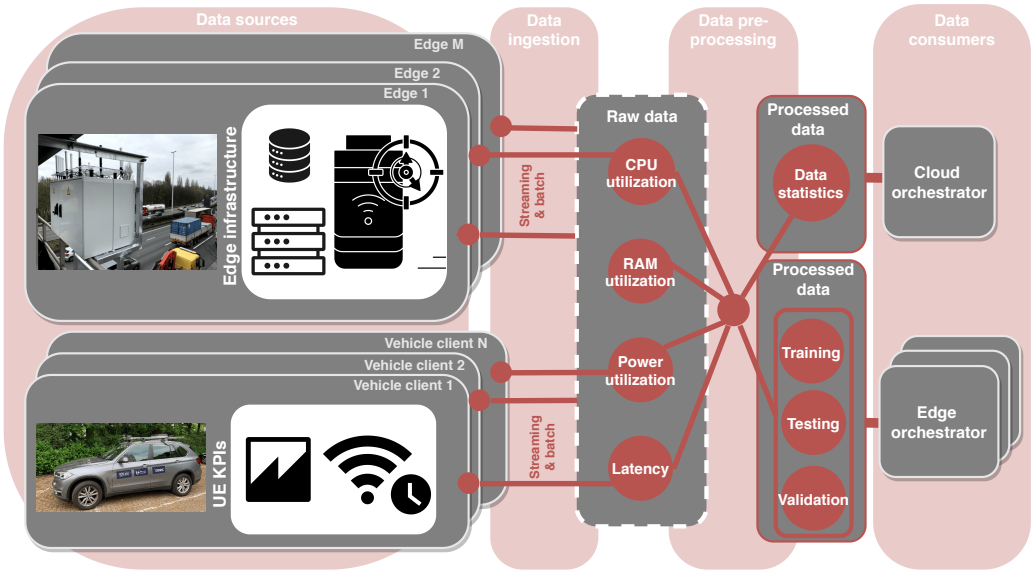


Figure 6.15: Data engineering pipeline in our PoC setup.

enforced from the orchestration layer), as decision-making process is distributed, taking into account KPIs measured at the user's side. One example of edge application that might benefit from intelligent orchestration is a BSA, used in our performance evaluation presented in Section 6.2.3.3.

The BSA type of edge V2X application is a containerized application used for creating topological in-advance area-specific notifications for vehicles based on the events that occurred behind them [197]. As discussed in detail in Chapter 5, the notifications are disseminated to different topological areas, and they contain instructions/warnings for the vehicles, while requiring some action from them to improve the driving conditions on the road, such as to clear the lane, to increase/decrease the speed, or to exit highway. These events can be either reported to edge application by specialized vehicles (e.g., emergency vehicles), or detected and reported by infrastructure sensors.

6.2.3.1 Data engineering pipeline

In Fig. 6.15 we illustrate the data engineering pipeline in our PoC deployment for experimentation with intelligent and automated edge orchestration. A crucial step towards enabling intelligence and automation is a robust data collection, which is then used for training, testing, and validation purposes.

To this end, our data engineering pipeline includes several types of data sources. In particular, we collect i) infrastructure metrics, i.e., CPU, memory, and power, utilization, and ii) network-related metrics, such as latency and bandwidth. Data is being ingested into message broker instances in each edge node, either as a stream or a batch. Such raw data is then processed by specialized helper services, i.e., MEC value-added services that perform data pre-processing, thereby making it suitable for further use. Data pre-processing includes

Algorithm 1: MAESTRO algorithm.

Result: Edge node selected for V2X application deployment
Edge application A_i deployed at the edge node N_k, orchestrated by Edge orchestrator E_j Start;
step 1; **while** *V2X application* A_i *is active* **do**
  Read KPI measurements for all edge nodes;
  Retrieve SVR model updates from Edge orchestrator O_j;
  Prepare CPU data for prediction of average response time;
  Predict $\overline{t}$ of A_i for all edge nodes E_k, $k \in (1, \dots N_E)$;
  if $\overline{t}$ *during* $\Delta T > t_{max}$ **then**
    Apply MCDM TOPSIS to make final decision for application relocation;
    Get decision;
    if *Application* A_i *is already deployed on the selected* E_k **then**
      go to step 1
    else
      Send notification about relocation to the source Edge orchestrator O_j;
      if *Edge orchestrator* O_j *accepts the decision* **then**
        Edge orchestrator O_j sends request for proactive application deployment to O_{j+1};
        if O_{j+1} *accepts the decision* **then**
          Deployment on E_{k+1} starts;
          The state/metadata is being transferred;
          O_j generates notification for vehicle edge client to reconnect from edge E_k to edge E_{k+1}
        else
          go to step 1, add flag to O_{j+1}
        end
      else
        go to step 1, add flag to O_j
      end
    end
  else
    go to step 1
  end
end

cleaning of the collected data, averaging, grouping into categories (e.g., CPU, power, average response time), performing statistical analysis, and packing it into datasets.

As illustrated in Fig. 6.15, processed data is made available for retrieving data statistics, which might be of interest for the cloud orchestrator (data consumer) to get insights into edge infrastructure and edge service performance. Importantly, data is also exposed for training, testing, and validation, so that edge orchestrators (data consumers) can consume the generated datasets to train their local ML models. This is especially important in distributed environments where data privacy is fundamental, so the training is performed locally.

Table 6.3: Parameters.

Parameter	Definition
A_i	V2X Application, $i \in \{1, \dots N_A\}$
N_A	Number of deployed V2X applications
O_j	Edge orchestrator, $j \in \{1, \dots N_O\}$
N_O	Number of Edge orchestrators
E_k	Edge node, $k \in \{1, N_E\}$
N_E	Number of edge nodes
t	Response time
$\overline{t}$	Average response time
t_{max}	maximum tolerable response time

6.2.3.2 Algorithm details

Here we briefly describe our hybrid edge application relocation algorithm, MAESTRO algorithm (Algorithm 1, parameters described in Table 6.3), which performs selection of a new target node to which the observed edge V2X application deployment should be relocated in order to maintain/achieve the required service response time. It works in an automated and intelligent way thanks to the MCDM mechanism that takes into account various metrics, such as CPU, memory, and power, utilization, as well as the predicted average response time for a vehicle client. The prediction is based on the SVR model that is trained at the edge orchestrator level. We use the TOPSIS class of MCDM algorithms, which is based on the comparison between all the alternatives included in the problem statement, and it is often used in solving large-scale decision-making problems in automotive industry [195]. On the other hand, we apply SVR, as a supervised learning technique, to find a function that approximates mapping from an input CPU load to average response time based on the training sample. Since edge orchestrators do not collect data from the other edge domains due to the security reasons, they cannot make decisions based on an extended perception that includes NFV infrastructure managed by other edge orchestrators. Therefore, the local SVR model, trained at the edge orchestrator level by using locally collected data, is then shared with the cloud orchestrator.

The cloud orchestrator predicts the average response time of edge V2X application for the next period of time ΔT. If predicted average response time does not exceed the t_{max}, which is the maximum tolerable response time for the edge application to provide a meaningful response (e.g., a credible record about the location and estimated time of arrival of the firetruck from behind) to the vehicle client, there is no need for relocation. The treshold t_{max} can be defined per application type, or even network slice type, so that orchestrators can correspondingly adjust their criteria. Also, this value should be low enough to enable proactive relocation, meaning that the average response time will not be degraded in the meantime while relocation is being performed. Such an automated and proactive approach is in line with the level 3/4 of autonomous networks proposed by ETSI in [177], which refer to automated distinction between different kinds of services, thereby analyzing the service performance and (proactively) adjusting the service based on the changing conditions in network and infrastructure.

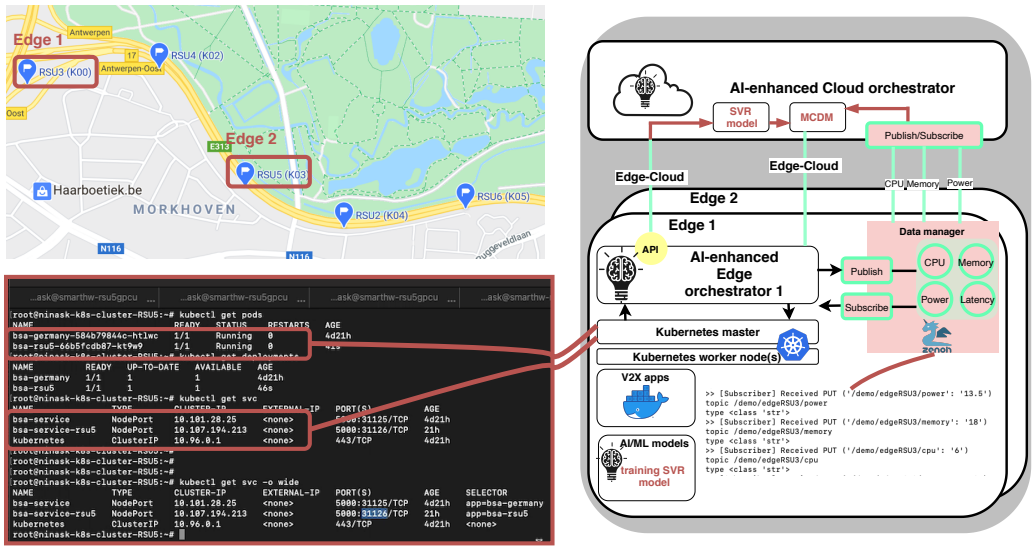
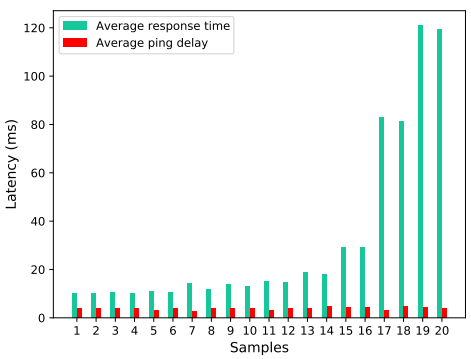


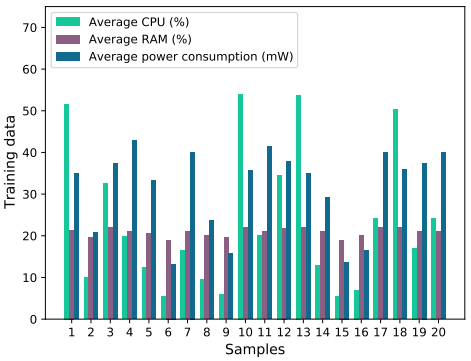
Figure 6.16: PoC utilized in Section 6.2.3.

If the decision is to relocate the edge application, the cloud orchestrator applies MCDM mechanism using the predicted value of average response time based on the CPU load for all edge domains, as well as other collected metrics (memory, power), in order to avoid relocating edge application to an edge node that e.g., experiences a high power consumption. Thus, at the same time, the cloud orchestrator is making a quality-aware decision, and trying to optimize the resource usage in all edge domains.

Following the steps provided in Algorithm 1, once the cloud orchestrator selects the edge orchestrator to be in charge of deploying relocated application instance, it sends the decision to the source edge orchestrator and triggers the relocation. If the edge orchestrator accepts this decision, it starts to proactively relocate the application to the selected target edge orchestrator. In case of stateful applications, whenever application instance is available at the target edge node, the transfer of state also needs to be performed before vehicle reconnects to it. This can be done by applying the container checkpoint and restore technique [108], which involves service downtime. Otherwise, if there is no state, but a certain metadata (e.g., location and speed of the firetruck) that will be used to configure the application, then it also needs to be transferred. Once the context and/or metadata are transferred, source edge orchestrator is sending notification to the vehicle client (as described in our previous work [198]) to change the endpoint of the edge application. The client on the vehicle side needs to be configured in the way to automatically process the notification from orchestrators, and to apply the rule of configuring service endpoints. Afterwards, vehicle is reconnected to the target edge application instance, which is orchestrated by the target edge orchestrators. Finally, in case any of the edge orchestrators do not accept the decision made by the cloud orchestrator (as described in Algorithm 1), it adds certain flags to those edge orchestrators. Such flags should be further studied by the cloud orchestrator to learn about the reasons for rejecting the decisions, which should also help to retrain and reconfigure ML models. This management of ML models is out of scope of this thesis, but part of our future work.

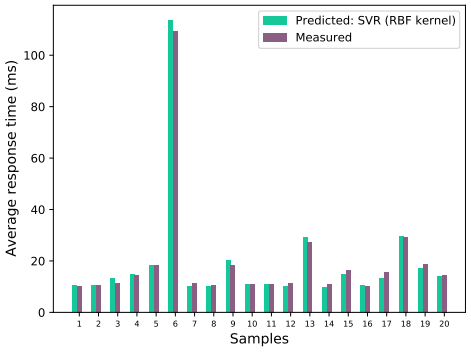


(a) Ping delay vs. Response time.

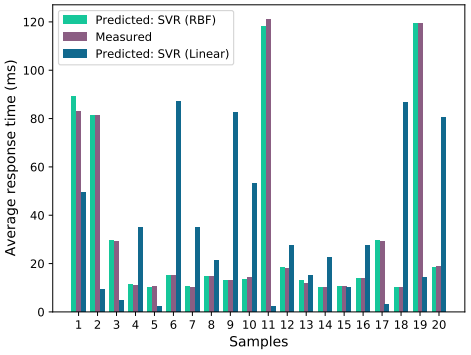


(b) Training data.

Figure 6.17: MAESTRO Results - part 1.

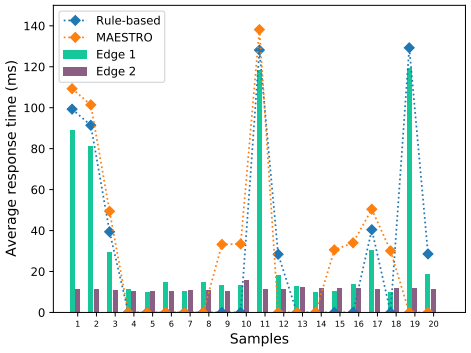


(a) Prediction based on testing data.

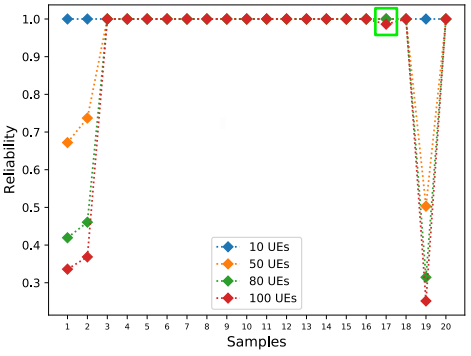


(b) Prediction based on training data.

Figure 6.18: MAESTRO Results - part 2.



(a) Gain achieved by relocation.



(b) Reliability.

Figure 6.19: MAESTRO Results - part 3.

Table 6.4: Results.

Model	R-squared	MSE	Average difference between predicted and measured data	Standard deviation	p-value in Kruskal Wallis test
SVR (RBF Kernel)	0.9979	2.64471	0.6651ms	1.484ms	0.9784
SVR (Linear Kernel)	0.8277	221.8706	9.985ms	11.7372ms	0.7251

6.2.3.3 Performance Analysis of MAESTRO

In the experimentation evaluation, we have utilized two MEC hosts from our PoC setup, i.e., *Edge 1* and *Edge 2*. To collect training data, we gradually stressed the edge V2X deployment on the RSU *Edge 1*. While performing the stress test at *Edge 1*, we have been collecting the response time measured at the client application in vehicle. The overall response time consists of communication (uplink and downlink) and processing/computational delay. If the average response time presented in Fig. 6.17a is observed, we can see how much are communication and computational delays contributing to the overall edge service response time. Samples indicate 20 batches of successive measurements, where each of the measurements lasted for one minute, and is represented by the mean value. The stress test in our scenario caused an increase in average response time, and as we can see in Fig. 6.17a, communication latency remains stable despite the stress test, thus, the computational latency on the edge node is affected.

In Fig. 6.17b, we show the average values of CPU load, RAM load, and power consumption, in the Kubernetes cluster at the *Edge 1*. Samples of measurements correspond to the samples of edge service response time in Fig. 6.17a. Given that scenario indicates a sporadic stress test from sample 1 to sample 20, in Fig. 6.17b we can notice the changes in the CPU load. Therefore, the goal is to explore the dependency of service quality experienced by user (i.e., vehicle) on the infrastructure metrics, such as CPU load. Based on the results of this data exploration on the collected metrics, we further exploit the dependency between the CPU load and the average response time to improve the service quality experienced by user (i.e., vehicle). Other collected metrics such as memory and power consumption will be still used by the MCDM algorithm to improve the final relocation (e.g., avoiding to use an edge node with high power consumption). As we collect both input (average CPU load) and output (average response time) data, we can apply any suitable supervised learning technique to determine the function of mapping input data to the expected output. In this experimentation setup, we used python[9] to apply two types of SVR depending on the kernel, i.e., Radial Basis Function (RBF) and Linear. Finally, we create two datasets, one for training, and another for testing.

The performance results are shown in Figures 6.17, 6.18, and 6.19, where Fig. 6.18b shows the prediction of an average response time based on the training data, and Fig. 6.18a the prediction based on the testing data. As we notice that SVR with RBF kernel produces larger R-squared value[10] (better fits the input to output), and lower MSE (determines the

[9]For this work, we have used the implementation of the SVR algorithm provided by the python library scikit-learn: `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html`

[10]R-squared is a coefficient of determination, a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model: `https://www.investopedia.com/terms/r/r-squared.asp`

accuracy of our model), this model is further used and applied in our algorithm for selecting the edge deployment.

As it can be noticed in Table 6.4, the SVR model achieves a high value of R-squared, i.e., 0.9979, and produces an MSE of 2.64471. The aforementioned result can be considered as a satisfactory level of prediction accuracy, given that average difference between predicted and measured data is less than 1 ms (0.6651 ms), which can be considered as negligible even for V2X applications such as BSA one that we used in this performance analysis.

For the type of V2X use cases where notifications/warnings are generated and collected from edge services (as a result of processing data from sensors and other vehicles), to extend the contextual perception of a vehicle, the result we obtained can be considered as satisfactory due to the following reasons. In case vehicle is moving with an average speed of 80 km/h, 15 ms can be considered as a tolerable latency for retrieving important warnings, as vehicle moves only for 0.33m until it gets a new notification. This of course needs to be studied with a more prominent attention in case of autonomous driving, or teleoperation of a vehicle, for which more ML models need to be studied and compared against each other to determine the satisfactory level of prediction accuracy. In Table 6.4, we also show the result of the Kruskal Wallis[11] test we applied to obtain a statistical significance of the difference between measured and predicted values of average response time. As $p-value$ is larger than 0.05, this result shows there is no statistically significant difference between measured and predicted data.

Finally, in Fig. 6.19a we show the result of the gain in average service response time that can be achieved by performing edge V2X service relocation in a proactive and automated way, i.e., by applying MAESTRO algorithm. First, the result shows the average response time measured at the client side for application instance running on *Edge 1*, and *Edge 2*. Second, it shows the behavior of MAESTRO algorithm against a simple rule-based algorithm, thereby examining the way they trigger service relocation. As the cloud orchestrator is constantly monitoring CPU data from different edge domains, it applies SVR model to predict the average response time for a particular type of edge V2X application. In case of MAESTRO, if predicted values of average response time in the upcoming three samples ($\Delta T = 1min$, three minutes upfront in total) is larger than t_{max}, which we consider as 15 ms for a used type of service, then the cloud orchestrator applies MCDM, and potentially requests an application relocation to *Edge 2* from *Edge 1*. On the other side, a rule-based algorithm simply compares the current average response time with the threshold (i.e., 15 ms), and triggers the relocation. In Fig. 6.19a, for each of the samples it can be seen whether these two algorithms trigger relocation for service deployment on the *Edge 1* or not. For instance, in sample 10, MAESTRO is triggering the service relocation from *Edge 1* to *Edge 2* in proactive way, which prevents the vehicle user to experience an increased response time, as in case of relocation the response time will be lower than 15ms, while on the contrary it will reach 120 ms on *Edge 1*. This exemplifies how MAESTRO is outperforming rule-based algorithms, which are most-commonly used in state-of-the-art NFV MANO systems.

However, we also need to check how these decisions affect the reliability of the service. As the service reliability can be defined as a ratio of served and received requests, in Fig. 6.19b

[11] The Kruskal Wallis test is one of the non-parametric tests that is used as a generalized form of the Mann Whitney U test: `https://www.statisticssolutions.com/kruskal-wallis-test/`.

we show how it changes from sample to sample in case service is placed on the *Edge 1*, and if multiple users (10, 50, 80, and 100) are consuming the service. The type of service we used in this performance analysis is capable of serving three concurrent requests, i.e., if concurrently served, they achieve average response time shown in Fig. 6.19a. In case of 89 ms response time (sample 1), the BSA application is capable of serving 33.59 requests/s (served). Clearly, the reliability of service will depend on the overall number of received requests, i.e., number of users. If there are 80 vehicles consuming the service at the same time (80 requests/s), the service reliability drops down to 0.42 in case service is consumed from *Edge 1*, which is completely unacceptable for most of the V2X services that require reliability of at least five nines (99.999%). Further, in sample 17, the reliability would drop to 0.9862 for 100 vehicles if service is not proactively relocated, which would happen in case of the rule-based algorithm as it does not proactively trigger the relocation in the 16th sample, as MAESTRO does. Same applies to sample 19, which brings completely intolerable reliability values if service is not previously relocated, as in case of MAESTRO being the one that triggers relocation in sample 18 (Fig. 6.19a). Such results show the true benefit of the quality-aware MAESTRO algorithm performing edge orchestration in a proactive and automated way, thereby re-attaching user from one edge to another when the algorithm triggers the relocation.

Concerning the re-attachment of vehicle client from one edge to another, in this experiment we utilized Zenoh framework to disseminate notifications from edge orchestrator to vehicle client. Furthermore, this client on the vehicle is capable of dynamically changing the service endpoint depending on the input received from edge orchestrators, by applying a new rule on its programmable data plane. As this concept is out of scope of this thesis, we leave it out for our future work. Also, in our future work, we plan to i) further extend the experimentation by adding more diversity to scenarios that can happen on the highways, thereby studying the impact of mobility, ii) study service relocation costs besides service reliability, and include them in the decision-making process, and iii) analyze and examine the efficiency of managing the decisions (that can be contradictory) made at the cloud and edge orchestration layers at different timescales.

6.3 Summary of the Chapter

In this Chapter, we studied the potential of applying AI/ML to edge orchestration solutions to automate and improve decision-making at the orchestration layer. Firstly, we proposed a closed-loop framework to automate orchestration operations, proposing the mechanisms on applying particular AI/ML techniques on orchestration processes, thereby evaluating the implications that could be brought by AI/ML (e.g., excessive resource consumption, demands for large datasets, and insufficient explainability). Secondly, we presented our attempts on designing ML-based algorithms that increase situational awareness of edge orchestrators, and thus, evaluate their impact on the service quality measured at the client side (e.g., vehicle).

Leveraging on the LSTM to forecast the resource consumption on the network edges, we proposed an optimized application context relocation mechanism that proactively relocates/migrates service/EdgeApp deployments from one edge to another in order to maintain

service continuity for moving users. With such an approach, we managed to achieve almost 100 ms lower end-to-end latency on average (measured at client side), if proactive service relocation is applied compared to a scenario with no relocation. This approach not only decreases latency but it also significantly lowers the fluctuations in latency measured at the client side, as standard deviation for no relocation results in 117 ms, with only 30 ms in case of relocation. Another algorithm that we designed and tested is MAESTRO, a hybrid algorithm based on SVR and MCDM. It shows i) negligible difference between measured and predicted data of the end-to-end latency (lower than 1 ms), ii) superiority of 87.5% or 110 ms lower average end-to-end latency than in case of simple rule-based mechanisms, and iii) 99.999% reliability for 100 vehicles simultaneously using service/EdgeApp, compared to 98.62% of rule based algorithms, which is not acceptable reliability for vehicular use cases (at least five nines required).

The performance analysis that we made in attempt to test applicability of simple AI/ML solutions for edge service orchestration, is conducted over the real-life proof-of-concept solutions that we built using the Smart Highway testbed. As there is in general a lack of testing results that involve real-life environments (most of the solutions are tested using simulations), we expect that such proof-of-concept will help us to further study impact of AI/ML on orchestration systems, and to collect even more meaningful results that will provide more insights into explainability of AI in case of network management and orchestration.

Chapter 7

Conclusion

The main objective of this research was to leverage on and seize the potential of the technologies such as 5G, MEC, and AI/ML, on the way towards creating an efficient and automated management and orchestration of services and resources across various edge domains, and achieving the low-latency-aware VNF placement and seamless migration of programmable services and EdgeApps.

This objective has been achieved throughout the four main contributions summarized and briefly discussed in Section 1.3. In this chapter, we summarize the main findings and conclusions of this thesis, mapping them to the aforementioned contributions. Finally, we present the future prospects of this research, and briefly discuss the new and exciting research directions spawned by the work on this thesis.

7.1 Main findings

Contribution 1: Management of virtualized and programmable networks: Surveys and performance evaluations

Benchmarking existing NFV MANO solutions In order to cope with strong heterogeneity in resources, services, vendors, etc., as well as high dynamicity in network traffics, followed by high mobility of users in vehicular communication nowadays, automation of network service management and orchestration can come up as a solution. As a study to exploit the features of network management and orchestration aiming to support delay sensitive applications, in Chapter 3 we presented the closed-loop life-cycle management of network services as an essential collaboration between orchestration, control, and monitoring. Furthermore, we created a comprehensive feature-based analysis of the most adopted existing MANO solutions. Finally, we extensively evaluated the performance of Open Baton and OSM, recognizing the main components of closed-loop life-cycle management in their MANO architectures. Having latency as a crucial parameter for all latency sensitive vehicular applications, we assessed the overall delay in service instantiation, in order to explore the contributing factor to overall latency that needs to be minimized. Regarding the latency requirements at the user equipment side, we further study the benefits of bringing CDNs to the

network edge by leveraging existing works and, in order to benchmark different MANO tools at the network edge, we measured the service instantiation delay of each solution. Based on the features and performance analysis of MANO tools, we presented valuable perspectives for incorporating MANO tools to realistic MEC-enhanced vehicular network scenarios. Taking into account both feature-based perspective and performance, our thorough analysis of OSM and Open Baton showed that Open Baton outperforms OSM in case of delay in instantiating CDNaaS instances. Furthermore, in the second experimentation setup where we compared different VIMs, our results show the impact of OpenStack and AWS on the performance of OSM, as well as the superiority of container-based service deployment over VM-based in case of Open Baton. For the edge network implementation in MEC, OSM performs better with OpenStack than AWS, due to the reasons presented above. However, the installation, configuration, and maintaining of OpenStack are unavoidable, and must be done by e.g., network administrators. As it can be seen in Fig. 3.15a and 3.15b, Docker outperforms OpenStack in terms of both OID and OTD. As containers are a lightweight solution comparing to VMs that we instantiated on top of the OpenStack, based on this result they prove to be more suitable for implementation on the resource-constrained network edge. As it was already stated in the Chapter, in case MANO systems decide to instantiate additional application instances to meet QoS and QoE requirements, it is important to obtain the values of overall instantiation and termination delays. Concerning values of these two metrics, expressed in the order of tens of seconds, we see that neither OSM Release 6 nor Open Baton Release 6 are ready to be used in a real deployment for vehicular networks, performing MANO of resources and services in MEC platforms. Potentially, in order to decrease the impact of such high delays on QoS, some predictions for service instantiation can be done in order to preempt the users' service requests. The results show that the impact of VIM is essential for the operation of MANO systems, since the same network services operating on top of NFVI managed by different VIMs take significantly more/less time to be instantiated/terminated. Although our results indicate that neither of these two MANO platforms has reached a level of maturity for a deployment in real vehicular networks with such VIM environments, the performance analysis and its construction as a repeatable testbench will serve to benchmark existing and future MANO solutions for MEC. From the conclusions presented above, we see that selecting a MANO tool is not a straightforward task, as different tools provide multiple benefits, depending on the perspective we take, which in our case was a MEC-enhanced vehicular communications perspective. The feature-based and performance analysis that we provided in this Chapter, are valuable for both academia and industry, and provide guidelines on facilitated incorporation of closed-loop life-cycle management in vehicular networks based on 5G and MEC. Additionally, having extensive feature-based and performance analysis presented in this chapter, our analysis can significantly facilitate development of new MANO tools.

Contribution 2: Resource and service orchestration for Connected Cooperative and Automated Mobility

The 5G ecosystem is comprised of the cellular 5G system along with a properly managed and orchestrated deployment of virtualized network and service functions in distributed cloud resources. Such ecosystem enables customized deployment and operation of services for different sectors of the vertical industry, and the automotive industry is a promising consumer

due to the high mobility and service demand with stringent QoS requirements. In Chapter 4, we proposed a solution for the orchestration of CCAM services within such 5G ecosystem to meet the stringent requirements of moving users, which connect to services in the network infrastructure. A key objective is the availability and continuity of low-latency services at the network infrastructure edges for a highly dynamic automotive scenario and the associated management and orchestration of these services in distributed edge clouds. Our proposed solution leverages a multi-tier orchestration system as well as localized management- and protocol operations for connected and collaborative edge resources. With the analytical and experimental evaluation, we draw conclusions on the gain in accelerating orchestration operations while balancing associated protocol and computational load over the distributed and multi-layered orchestration platforms. Considering the results, we signify the importance of the overall number of instances of reference points in the orchestration platform, which are established on-demand, and used as per MLA, because it reduces the number of hops for an orchestration request, thereby facilitating the access of edge-level orchestration entities to required resources for performing orchestration operations. Also, our results showed that the more instances of reference points are set up and authorized between edge-level orchestrators, the lower load is offloaded to the top-level orchestrators, which ultimately results in the decrease in their overall response time.

Contribution 3: Orchestrated Edge Network Applications (EdgeApps)

EdgeApps for 5G verticals In Chapter 5, we introduced Network applications, i.e., the so-called EdgeApps, which abstract the complexity of network infrastructure when it comes to providing vertical T&L services. As such, EdgeApps facilitate the deployments of those vertical services in real-life environments. The structure and behavior of EdgeApps is fluid, i.e., they can be designed and created on-demand to improve specific aspects of safety and efficiency in T&L operations through the delivery of vertical services, thereby contributing to prevent equipment collisions, to in-advance prepare for weather changes, and to identify unexpected movements of non-autonomous devices. As EdgeApps bind 5G and the vertical services, they are relevant for entrepreneurs, researchers, and the T&L industry, as tech entrepreneurs can use this concept to develop further case studies in the T&L and other sectors, research can use these outcomes to study the further improvements in 5G and beyond applications, and the T&L industry can benefit from a high-end contribution that details the role of 5G to tackle operational challenges.

Back-situation awareness application service As a continuation of the discussion on EdgeApps in Chapter 5, we introduced an on-demand 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. Due to the significant importance of decreasing the overall response time to the emergency events, we performed a thorough performance analysis of the BSA application service, measuring the impact of emergency on the MEC system resources, and service response time. Moreover, we introduced a metric called *panic indicator* that provides a notion

on 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 work, we see that it is important for 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 work, we derived 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.

Contribution 4: Intelligent and automated management and orchestration of services and resources

To alleviate the challenges in NFV MANO operations imposed mainly by manual interventions (i.e., delayed operations, reactive approach), there is a need to bring automation and intelligence to operations of orchestrating services and resources, especially the ones with stringent requirements for latency and capacity (e.g., V2X). In the last chapter of this thesis, we took several steps to enable an automated and intelligent MANO for B5G V2X systems. First, we studied and listed gaps in existing NFV MANO systems, and proposed a closed-loop framework to enable automation of the MANO process. Next, candidate AI/ML techniques are introduced for the considered NIFs and further challenges are elaborated. In Section 6.2.3, we presented and evaluated MAESTRO, an algorithm that makes proactive ML-driven decisions for edge service relocation in order to ensure QoS guarantees for V2X services. For the performance evaluation, we utilized the real-life testbeds, Smart Highway and Virtual Wall, and created a PoC for pursuing realistic experimentation and validation of

the impact that ML models have on the edge orchestration. We presented our efforts on improving MANO operation of service relocation towards achieving service continuity and required service quality, by applying an ML-based quality-aware concept that automates service relocation, thereby minimizing average response time and maximizing service reliability.

7.2 Future prospects of this research

Beyond 5G networks such as 6G will be built on top of the fully autonomous networks, with management capabilities such as self-configuration, self-healing, self-optimizing, and self-evolving, aspects that are not supported by current networks that largely depend on the automated assistance [199]. To successfully implement and deploy such autonomous networks, various innovative contributions are required in the area of network management, leveraging the solid experience and advancements in AI/ML during the last 5-10 years, which provided a new set of algorithms and tools to solve challenging problems in multiple domains by learning complex relationships directly from the data. However, one of the segments in 6G that will need a deep transformation towards the adoption of NI is the service and resource orchestration layer, as the traditional approaches in orchestration urge for advancements that will make them able to swiftly respond to not only an influx of collected data and service requests but also to take advantage of the intelligence that will be deployed in the network itself. In line with our final contribution in this PhD thesis, enabling automatic and service-agnostic management and orchestration opens up various new research directions (illustrated in Fig. 7.1), which will be focus of our future work. Such future MANO requires creation of innovative architecture of loosely coupled management and orchestration elements, with open and programmable interfaces that will enable the communication between those elements and NIFs running inside the distributed network segments. The NIFs realized as diverse AI/ML algorithms will feed the orchestration elements either with the data collected from the distributed network segments, or with the decisions that will enhance the performance of each orchestration element in the collaborative orchestration system. The NIFs collect the data from various sources in the network (performance metrics measured at the client side, edge resource consumption, latency, bandwidth, user mobility, etc.), and make predictions and decisions that will help MANO orchestrators towards improving their operations, adding the cognition required to remove or reduce the need of the human-in-the-loop and realizing a complete closed-loop network management approach, while being able to deploy NIFs across all network segments and taking care of their conflicting decisions.

In particular, to solve a particular task with the help of AI/ML, engineers and researchers are usually building a single robust ML model, which is trained, tested, and validated, using large datasets. However, when it comes to more complex tasks, such as management and orchestration of network services, it is impractical to automate it by applying a single AI/ML technique due to the network complexity, i.e., creating a single model that will consider all kinds of data sources and impact factors from the network becomes unfeasible (one size does not fit all). Instead, NIFs are paving the way toward distributing the network intelligence, so that MANO elements can combine and harmonize outputs (decisions and predictions) from various NIFs what focus on a particular task (e.g., mobility pattern, resource utilization, anomaly detection in service operation). To do so, NIFs can be injected at different segments in the network ecosystem, building future prospects of injecting AI into beyond

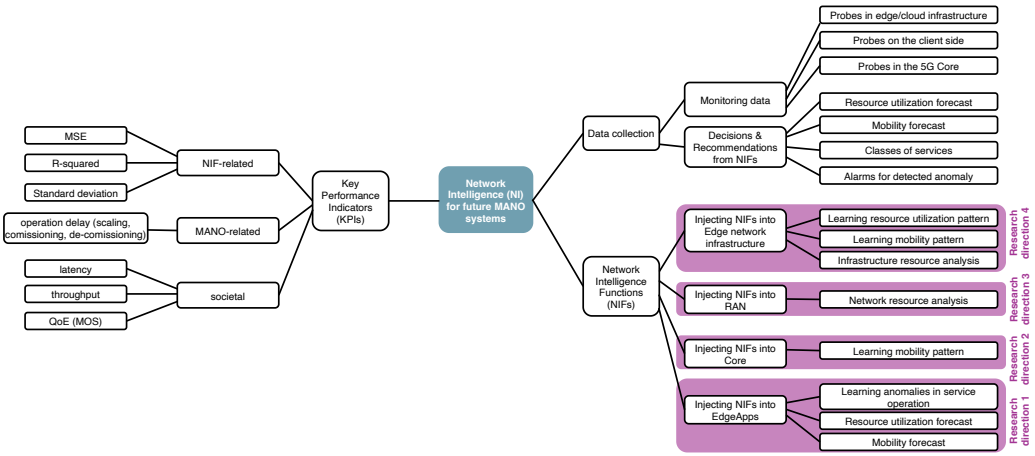


Figure 7.1: Future Management and Orchestration: The overview of aspects relevant for bringing intelligence in terms of NIFs to future service orchestration.

5G orchestration systems, as illustrated in Fig. 7.1. Thus, following the work executed by standardization bodies, such as ETSI, in the two working groups: ZSM, and ENI, the research executed in the fourth contribution of this thesis, and the future research directions that we presented in this section, contribute to accelerating the automatic execution of MANO operations by injecting the intelligence into various network segments, such as edge services/EdgeApps, orchestrators, platform managers, radio, and core network.

Such perspective is even going beyond the scope of current standardization activities in these two working groups, which opens up the potential of extending the standardization activities, and thus contributing to standardization bodies. The performance of such optimized and automated fully-fledged orchestration system will be measured through the three different groups of KPIs, the ones measuring the performance of AI/ML models deployed in the NIFs, the ones evaluating the performance of MANO operations through measuring their delays, and finally, also determining the ones that directly impact the users' perception, i.e., QoS and QoE.

Bibliography

[1] 3GPP, "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows," *3GPP TS 23.434*.

[2] T. Frisanco, P. Tafertshofer, P. Lurin, and R. Ang, "Infrastructure Sharing for Mobile Network Operators," *Conference proceedings of The International Conference on Information Networking*, pp. 444–448, 2008, doi: http://dx.doi.org/10.1109/ICOIN.2008.4472768.

[3] Cisco and I. Cisco Systems, "Cisco Visual Networking Index: Forecast and Trends, 2018-2023," [Online] Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[4] 5G-ACIA, "5G for Connected Industries and Automation (White Paper - Second Edition)," no. November, p. 28, 2018, [Online] Available: https://bit.ly/2BGSMLA.

[5] J. Van De Belt, H. Ahmadi, and L. E. Doyle, "Defining and Surveying Wireless Link Virtualization and Wireless Network Virtualization," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1603–1627, 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2704899.

[6] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," pp. 32–39, 2016, doi: http://dx.doi.org/10.1109/MCOM.2016.7514161.

[7] D. Zhang, *Virtual Resource-Sharing Mechanisms in Software-Defined and Virtualized Wireless Network*, 2018, [Online] Available: https://bit.ly/2UJXa6K.

[8] E. Coronado, Z. Yousaf, and R. Riggio, "LightEdge: Mapping the Evolution of Multi-Access Edge Computing in Cellular Networks," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 24–30, 2020, doi: http://dx.doi.org/10.1109/MCOM.001.1900690.

[9] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018, doi: http://dx.doi.org/10.1109/MCOMSTD.2018.1800013.

[10] A. Reznik. Multi-access Edge Computing (MEC). [Online] Available: https://www.etsi.org/technologies/multi-access-edge-computing?jjj=1573496932482.

[11] R. Vilalta, R. Munoz, R. Casellas, and R. Martinez, "Dynamic deployment of virtual GMPLS-controlled elastic optical networks using a virtual network resource broker on the ADRENALINE testbed," *International Conference on Transparent Optical Networks*, pp. 2–5, 2013, doi: http://dx.doi.org/10.1109/ICTON.2013.6602843.

[12] F. Slyne, R. Giller, J. Singh, and M. Ruffini, "Experimental Demonstration of DPDK Optimised VNF Implementation of Virtual DBA in a Multi-Tenant PON," pp. 1–3, Sep. 2018, doi: http://dx.doi.org/10.1109/ECOC.2018.8535109.

[13] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An sdn/nfv based framework for management and deployment of service based 5g core network," *China Communications*, vol. 15, no. 10, pp. 86–98, 2018, doi: http://dx.doi.org/10.1109/CC.2018.8485472.

[14] M. Jiang, D. Xenakis, S. Costanzo, N. Passas, and T. Mahmoodi, "Radio Resource Sharing as a service in 5G: A software-defined networking approach," *Computer Communications*, vol. 107, pp. 13–29, 2017, doi: http://dx.doi.org/10.1016/j.comcom.2017.03.006.

[15] O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Spectrum Sharing in Multi-Tenant 5G Cellular Networks: Modeling and Planning," *IEEE Access*, vol. 7, pp. 1602–1616, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2018.2886447.

[16] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 358–380, 2015, doi: http://dx.doi.org/10.1109/COMST.2014.2352118.

[17] Balasubramanian, Chandrasekar and Ramanujam, Suresh, "Software defined networking (sdn)," 2019, online [Available]: https://www.gavstech.com/software-defined-networking-sdn/, Last accessed on 2022-6-11.

[18] A. Kliks, B. Musznicki, K. Kowalik, and P. Kryszkiewicz, "Perspectives for resource sharing in 5G networks," *Telecommunication Systems*, vol. 68, no. 4, pp. 605–619, 2018, doi: http://dx.doi.org/10.1007/s11235-017-0411-3. [Online]. Available: https://doi.org/http://dx.doi.org/10.1007/s11235-017-0411-3

[19] G. Liu, Y. Huang, F. Wang, J. Liu, and Q. Wang, "5G features from operation perspective and fundamental performance validation by field trial," *China Communications*, vol. 15, no. 11, pp. 33–50, 2018, doi: http://dx.doi.org/10.1109/CC.2018.8543047.

[20] J. K. A. et al., "Contributors," in *Philosophy of Technology and Engineering Sciences*, ser. Handbook of the Philosophy of Science, A. Meijers, Ed. Amsterdam: North-Holland, 2009, pp. vii–xi, doi: http://dx.doi.org/10.1016/B978-0-444-51667-1.50003-3.

[21] ETSI, "Network Functions Virtualisation (NFV) Release 2; Testing; NFVI Compute and Network Metrics Specification," *ETSI GS NFV-TST 008 v2.4.1*, 2018, online [Available]: https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/008/02.04.01_60/gs_nfv-tst008v020401p.pdf.

[22] F. Spinelli and V. Mancuso, "Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility," *IEEE Communications*

Surveys Tutorials, vol. 23, no. 1, pp. 596–630, 2021, doi: https://doi.org/10.1109/COMST.2020.3037674.

[23] 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.

[24] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, Jan 2018, doi: http://dx.doi.org/10.1109/MCOM.2018.1700467.

[25] M. Amadeo, C. Campolo, A. Molinaro, J. Harri, C. E. Rothenberg, and A. Vinel, "Enhancing the 3GPP V2X Architecture with Information-Centric Networking," *Future Internet*, vol. 11, no. 9, 2019, doi: http://dx.doi.org/10.3390/fi11090199. [Online]. Available: https://www.mdpi.com/1999-5903/11/9/199

[26] 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.

[27] Z. Laaroussi, R. Morabito, and T. Taleb, "Service Provisioning in Vehicular Networks Through Edge and Cloud: An Empirical Analysis," pp. 1–6, Oct 2018, doi: http://dx.doi.org/10.1109/CSCN.2018.8581855.

[28] Z. Ning and X. Wang, "Mobile Edge Computing-Enabled 5G Vehicular Networks: Toward the Integration of Communication and Computing," *IEEE Vehicular Technology Magazine*, vol. 14, no. March, pp. 54–61, 2019, doi: http://dx.doi.org/10.1109/MVT.2018.2882873.

[29] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, Aug 2019, doi: http://dx.doi.org/10.1109/TVT.2019.2917890.

[30] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation Offloading and Resource Allocation in Vehicular Networks Based on Dual-Side Cost Minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, Feb 2019, doi: http://dx.doi.org/10.1109/TVT.2018.2883156.

[31] V. H. Hoang, T. M. Ho, and L. B. Le, "Mobility-aware Computation Offloading in MEC based Vehicular Wireless Networks," *IEEE Communications Letters*, pp. 1–1, 2019, doi: http://dx.doi.org/10.1109/LCOMM.2019.2956514.

[32] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. S. Quek, "Computation Offloading for Mobile Edge Computing Enabled Vehicular Networks," *IEEE Access*, vol. 7, pp. 62 624–62 632, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2019.2915959.

[33] T. Taleb, S. Member, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," vol. 19, no. 3, pp. 1657–1681, 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2705720.

[34] R. Soua, I. Turcanu, F. Adamsky, D. Führer, and T. Engel, "Multi-Access Edge Computing for Vehicular Networks: A Position Paper," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–6, doi: http://dx.doi.org/10.1109/GLOCOMW.2018.8644392.

[35] N. Slamnik-Kriještorac, H. Kremo, M. Ruffini, and J. M. Marquez-Barja, "Sharing Distributed and Heterogeneous Resources toward End-to-End 5G networks: A Comprehensive Survey and a Taxonomy," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2020, doi: http://dx.doi.org/10.1109/COMST.2020.3003818.

[36] A. Abdelaziz, A. Fong, A. Gani, S. Khan, F. Alotaibi, and M. Khan, "On Software-Defined Wireless Network (SDWN) Network Virtualization: Challenges and Open Issues," *Computer Journal*, vol. 60, pp. 1510–1519, 10 2017, doi: http://dx.doi.org/10.1093/comjnl/bxx063.

[37] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, July 2017, doi: http://dx.doi.org/10.1109/MCOM.2017.1601150.

[38] T. Soenen, W. Tavernier, M. Peuster, F. Vicens, G. Xilouris, S. Kolometsos, M. Kourtis, and D. Colle, "Empowering Network Service Developers: Enhanced NFV DevOps and Programmable MANO," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 89–95, May 2019, doi: http://dx.doi.org/110.1109/MCOM.2019.1800810.

[39] N. F. Saraiva de Sousa, D. A. Lachos Perez, R. V. Rosa, M. A. Santos, and C. Esteve Rothenberg, "Network Service Orchestration: A Survey," *Computer Communications*, vol. 142-143, p. 69–94, Jun 2019, doi: http://dx.doi.org/10.1016/j.comcom.2019.04.008. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2019.04.008

[40] A. H. Celdrán, G. Clemente, and G. M. Pérez, "Automatic Monitoring Management for 5G Mobile Networks," 2017, doi: https://doi.org/10.1016/j.procs.2017.06.102.

[41] O. Zhdanenko, J. Liu, R. Torre, S. Mudriievskiy, H. Salah, G. T. Nguyen, and F. H. P. Fitzek, "Demonstration of Mobile Edge Cloud for 5G Connected Cars," *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–2, 2019, doi: http://dx.doi.org/10.1109/CCNC.2019.8651783.

[42] M. Peuster, M. Marchetti, G. García de Blas, and H. Karl, "Automated testing of NFV orchestrators against carrier-grade multi-PoP scenarios using emulation-based smoke testing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 172, Jun 2019, doi: http://dx.doi.org/10.1186/s13638-019-1493-2.

[43] G. M. Yilma, F. Z. Yousaf, V. Sciancalepore, and X. Costa-Perez, "On the Challenges and KPIs for Benchmarking Open-Source NFV MANO Systems: OSM vs ONAP," pp. 1–6, Dec 2019, online [Available]: https://arxiv.org/ftp/arxiv/papers/1904/1904.10697.pdf.

[44] T. V. Doan, G. T. Nguyen, H. Salah, S. Pandi, M. Jarschel, R. Pries, and F. H. P. Fitzek, "Containers vs Virtual Machines: Choosing the Right Virtualization Technology for Mobile Edge Cloud," in *2019 IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 46–52, doi: https://doi.org/10.1109/5GWF.2019.8911715.

[45] T. Salah, M. J. Zemerly, C. Y. Yeun, M. Al-Qutayri, and Y. Al-Hammadi, "Performance Comparison between Container-based and VM-based Services," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, March 2017, pp. 185–190, doi: https://doi.org/10.1109/ICIN.2017.7899408.

[46] T. Sechkova, M. Paolino, and D. Raho, "Virtualized Infrastructure Managers for Edge Computing: OpenVIM and OpenStack Comparison," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2018, pp. 1–6, doi: https://doi.org/10.1109/BMSB.2018.8436858.

[47] N. Slamnik-Kriještorac, E. de Britto e Silva, E. Municio, H. Carvalho de Resende, S. Hadiwardoyo, and J. Marquez-Barja, "Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context," *Sensors 2020*, vol. 20, 2020, doi: https://doi.org/10.3390/s20143852.

[48] G. M. Yilma, Z. F. Yousaf, V. Sciancalepore, and X. Costa-Perez, "Benchmarking open source NFV MANO systems: OSM and ONAP," *Computer Communications*, vol. 161, pp. 86 – 98, 2020, doi: https://doi.org/10.1016/j.comcom.2020.07.013.

[49] P. Trakadas, P. Karkazis, H. C. Leligou, T. Zahariadis, F. Vicens, A. Zurita, P. Alemany, T. Soenen, C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, and D. Kyriazis, "Comparison of Management and Orchestration Solutions for the 5G Era," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, 2020, doi: http://dx.doi.org/10.3390/jsan9010004. [Online]. Available: https://www.mdpi.com/2224-2708/9/1/4

[50] 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.

[51] G. Baggio, A. Francescon, and R. Fedrizzi, "Multi-domain service orchestration with X-MANO," in *2017 IEEE Conference on Network Softwarization NetSoft)*, 2017, pp. 1–2, doi: http://dx.doi.org/10.1109/NETSOFT.2017.8004259.

[52] B. Sonkoly, J. Czentye, R. Szabó, D. Jocha, J. Elek, S. Sahhaf, W. Tavernier, and F. Risso, "Multi-Domain Service Orchestration Over Networks and Clouds: A Unified Approach," *Computer Communication Review*, vol. 45, pp. 377–378, 2015, doi: http://dx.doi.org/10.1145/2829988.2790041.

[53] J. F. Riera, J. Batallé, J. Bonnet, M. Días, M. McGrath, G. Petralia, F. Liberati, A. Giuseppi, A. Pietrabissa, A. Ceselli, A. Petrini, M. Trubian, P. Papadimitrou, D. Dietrich, A. Ramos, J. Melián, G. Xilouris, A. Kourtis, T. Kourtis, and E. K. Markakis, "TeNOR: Steps towards an orchestration platform for multi-PoP NFV deployment," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 2016, pp. 243–250, doi: http://dx.doi.org/10.1109/NETSOFT.2016.7502419.

[54] 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: https://doi.org/10.1109/TVT.2016.2591558.

[55] 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: https://doi.org/10.1109/MVT.2016.2641139.

[56] 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)*, 2019, pp. 1–7, doi: https://doi.org/10.1109/PIMRC.2019.8904431.

[57] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A Tutorial on 5G NR V2X Communications," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021, doi: http://dx.doi.org/10.1109/COMST.2021.3057017.

[58] M. Amadeo, C. Campolo, A. Molinaro, J. Harri, C. E. Rothenberg, and A. Vinel, "Enhancing the 3GPP V2X architecture with information-centric networking," *Future Internet, Vol.11, N°9, 19 September 2019*, 09 2019, doi: https://doi.org/10.3390/fi11090199. [Online]. Available: http://www.eurecom.fr/publication/6025

[59] N. Cardona, E. Coronado, S. Latré, R. Riggio, and J. M. Marquez-Barja, "Software-Defined Vehicular Networking: Opportunities and Challenges," *IEEE Access*, vol. 8, pp. 219 971–219 995, 2020, doi: http://dx.doi.org/10.1109/ACCESS.2020.3042717.

[60] 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: https://doi.org/10.1109/MVT.2017.2752798.

[61] F. Malandrino and C.-F. Chiasserini, "Present-day verticals and where to find them: A data-driven study on the transition to 5G," in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2018, pp. 25–28, doi: https://doi.org/10.23919/WONS.2018.8311657.

[62] C. Patachia-Sultanoiu, I. Bogdan, G. Suciu, A. Vulpe, O. Badita, and B. Rusti, "Advanced 5G Architectures for Future NetApps and Verticals," in *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2021, pp. 1–6, doi: https://doi.org/10.1109/BlackSeaCom52164.2021.9527889.

[63] A. Bonea, C. Patachia-Sultanoiu, M. Iordache, I. Constantin, A. Radulescu, C. R. Comsa, and C. F. Caruntu, "Automated Onboarding, Testing and Validation Framework for NetApps," in *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2021, pp. 1–4, doi: https://doi.org/10.1109/ISSCS52333.2021.9497420.

[64] L. Boero, R. Bruschi, F. Davoli, M. Marchese, and F. Patrone, "Satellite Networking Integration in the 5G Ecosystem: Research Trends and Open Challenges," *IEEE Network*, vol. 32, no. 5, pp. 9–15, 2018, doi: https://doi.org/10.1109/MNET.2018.1800052.

[65] A. Fornes-Leal, R. Gonzalez-Usach, C. E. Palau, M. Esteve, D. Lioprasitis, A. Priovolos, G. Gardikis, S. Pantazis, S. Costicoglou, A. Perentos, E. Hadjioannou, M. Georgiades, and A. Phinikarides, "Deployment of 5G Experiments on Underserved Areas using the Open5GENESIS Suite," in *2021 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2021, pp. 1–4, doi: https://doi.org/10.1109/SmartNets50376.2021.9555428.

[66] K. C. Apostolakis, G. Margetis, C. Stephanidis, J.-M. Duquerrois, L. Drouglazet, A. Lallet, S. Delmas, L. Cordeiro, A. Gomes, M. Amor, A. D. Zayas, C. Verikoukis, K. Ramantas, and I. Markopoulos, "Cloud-Native 5G Infrastructure and Network Applications (NetApps) for Public Protection and Disaster Relief: The 5G-EPICENTRE Project," in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 235–240, doi: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482425.

[67] K. Trichias, G. Landi, E. Seder, J. Marquez-Barja, R. Frizzell, M. Iordache, and P. Demestichas, "VITAL-5G: Innovative Network Applications (NetApps) Support over 5G Connectivity for the Transport & Logistics Vertical," in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 437–442, doi: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482437.

[68] S. P. Shah, B. J. Pattan, N. Gupta, N. D. Tangudu, and S. Chitturi, "Service Enabler Layer for 5G Verticals," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 269–274, doi: https://doi.org/10.1109/5GWF49715.2020.9221425.

[69] P. Du, A. Nakao, L. Zhong, J. Ma, and R. Onishi, "Service-aware 5G/B5G Cellular Networks for Future Connected Vehicles," in *2021 IEEE International Smart Cities Conference (ISC2)*, 2021, pp. 1–4, doi: https://doi.org/10.1109/ISC253183.2021.9562863.

[70] R. Aringhieri, G. Carello, and D. Morale, "Ambulance location through optimization and simulation: the case of Milano urban area," in *XXXVIII Annual Conference of the Italian Operations Research Society Optimization and Decision Sciences:1-29*, 2007, online [Available]: https://www.semanticscholar.org/paper/Ambulance-location-through-optimization-and-%3A-the-Aringhieri-Carello/183cf61870c90f78b76a5aac96d31906fc13ef7b.

[71] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457509003261

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

[73] 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, no. 2, pp. 528–542, June 2009, doi: https://doi.org/10.1016/j.ejor.2008.02.003.

[74] 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.

[75] 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.

[76] M. Siegel, "Emergency Vehicle Alert System," *U.S. Patent 6,958,707.*, 2005, online [Available]: https://patents.google.com/patent/US20030043056A1/en.

[77] 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.

[78] 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.

[79] 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)*, 2019, pp. 88–92, doi: http://dx.doi.org/10.1109/COGSIMA.2019.8724309.

[80] 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)*, 2015, pp. 1–5, doi: http://dx.doi.org/10.1109/NTMS.2015.7266466.

[81] N. Slamnik-Kriještorac, H. C. Carvalho de Resende, C. Donato, S. Latré, R. Riggio, and J. Marquez-Barja, "Leveraging Mobile Edge Computing to Improve Vehicular Communications," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–4, doi: http://dx.doi.org/10.1109/CCNC46108.2020.9045698.

[82] 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.

[83] 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)*, 2019, pp. 1–7, doi: http://dx.doi.org/10.1109/PIMRC.2019.8904431.

[84] 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)*, 2019, pp. 1375–1379, doi: http://dx.doi.org/10.1109/ICCT46805.2019.8947060.

[85] R. Halili, F. Z. Yousaf, N. Slamnik-Krieštorac, G. M. Yilma, M. Liebsch, E. de Britto e Silva, S. A. Hadiwardoyo, R. Berkvens, and M. Weyn, "Leveraging MEC in a 5G System for Enhanced Back Situation Awareness," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020, pp. 309–320, doi: http://dx.doi.org/10.1109/LCN48667.2020.9314838.

[86] 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*, vol. 24, no. 9, pp. 665–668, 2007, doi: http://dx.doi.org/10.1136/emj.2007.047654.

[87] J. Pell, J. Sirel, A. Marsden, I. Ford, and S. Cobbe, "Effect of Reducing Ambulance Response Times on Deaths from out of Hospital Cardiac Arrest: Cohort Study," *BMJ (Clinical research ed.)*, vol. 322, pp. 1385–8, 07 2001, doi: http://dx.doi.org/10.1136/bmj.322.7299.1385.

[88] 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.

[89] N. Kapileswar, P. V. Santhi, V. K. R. Chenchela, and C. H. V. S. Prasad, "A Fast Information Dissemination System for Emergency Services over Vehicular Ad Hoc Networks," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 236–241, doi: http://dx.doi.org/10.1109/ICECDS.2017.8389862.

[90] C.-H. Hong and B. Varghese, "Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms," *ACM Comput. Surv.*, vol. 52, no. 5, Sep. 2019, doi: https://dx.doi.org/10.1145/3326066. [Online]. Available: https://doi.org/10.1145/3326066

[91] A. Zomaya, "Keynote 2: Resource Management in Edge Computing: Opportunities and Open Issues," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–1, doi: http://dx.doi.org/10.1109/ISCC47284.2019.8969601.

[92] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource Management Techniques for Cloud/Fog and Edge Computing: An Evaluation Framework and Classification," *Sensors*, vol. 21, no. 5, 2021, doi: http://dx.doi.org/10.3390/s21051832. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1832

[93] S. Fu, F. Yang, and Y. Xiao, "AI Inspired Intelligent Resource Management in Future Wireless Network," *IEEE Access*, vol. 8, pp. 22 425–22 433, 2020, doi: http://dx.doi.org/10.1109/ACCESS.2020.2968554.

[94] ETSI, "Artificial Intelligence and future directions for ETSI," *ETSI White Paper No. 34*, no. 34, 2020, online [Available]: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp34_Artificial_Intellignce_and_future_directions_for_ETSI.pdf.

[95] ——, "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Report on enabling autonomous management in NFV-MANO," *ETSI GR NFV-IFA 041*, 2021, online [Available]: https://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/041/04.01.01_60/gr_NFV-IFA041v040101p.pdf.

[96] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, "Machine learning-based zero-touch network and service management: A survey," *Digital Communications and Networks*, 2021, doi: https://dx.doi.org/10.1016/j.dcan.2021.09.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864821000614

[97] ETSI, "ETSI Zero-touch Network and Service Management (ZSM)," 2022.

[98] ETSI, "Experiential Networked Intelligence (ENI); Overview of Prominent Control Loop Architectures," *ETSI GR ENI 017 V2.1.1*, 2021, online [Available]: https://www.etsi.org/deliver/etsi_gr/ENI/001_099/017/02.01.01_60/gr_ENI017v020101p.pdf.

[99] ETSI, "ETSI Experiential Networked Intelligence," 2022.

[100] J. Guo, B. Song, Y. He, F. R. Yu, and M. Sookhak, "A Survey on Compressed Sensing in Vehicular Infotainment Systems," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2662–2680, Fourthquarter 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2705027.

[101] R. C. Abeywardana, K. W. Sowerby, and S. M. Berber, "Empowering Infotainment Applications: A Multi-Channel Service Management Framework for Cognitive Radio Enabled Vehicular Ad Hoc Networks," pp. 1–5, June 2018, doi: http://dx.doi.org/10.1109/VTCSpring.2018.8417749.

[102] K. Su, Y. Mo, L. Chen, W. Chang, W. Hu, C. Yu, and J. Tang, "An In-Vehicle Infotainment Platform for Integrating Heterogeneous Networks Interconnection," pp. 1–2, May 2018, doi: http://dx.doi.org/10.1109/ICCE-China.2018.8448834.

[103] Y. Chen and W. Liao, "Mobility-Aware Service Function Chaining in 5G Wireless Networks with Mobile Edge Computing," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6, doi: http://dx.doi.org/10.1109/ICC.2019.8761306.

[104] B. Blanco, J. Oscar, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino, E. Sfakianakis, F. Liberal, and G. Xilouris, "Computer Standards & Interfaces Technology pillars in the architecture of future 5G mobile networks : NFV , MEC and SDN," *Computer Standards & Interfaces*, vol. 54, no. January, pp. 216–228, 2017, doi: http://dx.doi.org/10.1016/j.csi.2016.12.007. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2016.12.007

[105] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 303–324, 2017, doi: http://dx.doi.org/10.1109/COMST.2016.2597193.

[106] ETSI, "Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines," *ETSI GS MEC-IEG 006 V1.1.1*, 2017, online [Available]: https://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/006/01.01.01_60/gs_MEC-IEG006v010101p.pdf.

[107] A. Reznik. Open Source MANO. [Online] Available: https://www.etsi.org/technologies/nfv/open-source-mano.

[108] Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, "Migration modeling and learning algorithms for containers in fog computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 712–725, 2019, doi: https://doi.org/10.1109/TSC.2018.2827070.

[109] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A Survey on Service Migration in Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 23 511–23 528, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2828102.

[110] R. A. Addad, D. L. Cadette Dutra, M. Bagaa, T. Taleb, and H. Flinck, "Towards a Fast Service Migration in 5G," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–6, doi: http://dx.doi.org/10.1109/CSCN.2018.8581836.

[111] H. Abdah, J. P. Barraca, and R. L. Aguiar, "QoS-Aware Service Continuity in the Virtualized Edge," *IEEE Access*, vol. 7, pp. 51 570–51 588, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2019.2907457.

[112] M. Horii, Y. Kojima, and K. Fukuda, "Stateful process migration for edge computing applications," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6, doi: http://dx.doi.org/10.1109/WCNC.2018.8377072.

[113] A. Strunk, "Costs of virtual machine live migration: A survey," in *2012 IEEE Eighth World Congress on Services*, 2012, pp. 323–329, doi: http://dx.doi.org/10.1109/SERVICES.2012.23.

[114] H. Maziku and S. Shetty, "Towards a network aware vm migration: Evaluating the cost of vm migration in cloud data centers," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014, pp. 114–119, doi: http://dx.doi.org/10.1109/CloudNet.2014.6968978.

[115] W. Hamid and M. A. Shah, "AWS Support in Open Source Mano Monitoring Module," pp. 1–6, Sep. 2018, doi: http://dx.doi.org/10.23919/IConAC.2018.8749021.

[116] 5G Tango, "5G Tango project description, outcomes, and objectives," 2020, https://www.5gtango.eu/about-5g-tango/, Last accessed on 2020-6-15.

[117] G. 5GTANGO Consortium. A brief overview of monitoring solutions embraced by 5G MANOs. [Online] Available: https://www.5gtango.eu/blog/58-a-brief-overview-of-monitoring-solutions-embraced-by-5g-manos.html.

[118] OpenStack, "Openstack official documentation," 2020, https://www.openstack.org/, Last accessed on 2020-6-15.

[119] Amazon Web Services, "Amazon web services (aws) official documentation," 2020, https://aws.amazon.com/, Last accessed on 2020-6-15.

[120] VMWare, "Vmware official documentation," 2020, https://www.vmware.com/, Last accessed on 2020-6-15.

[121] OpenVIM, "Openvim official documentation," 2020, https://www.openvim.com/, Last accessed on 2020-6-15.

[122] W. Hamid and M. A. Shah, "AWS Support in Open Source Mano Monitoring Module," pp. 1–6, Sep. 2018, doi: http://dx.doi.org/10.23919/IConAC.2018.8749021.

[123] P. Humphrey. Understanding When to use RabbitMQ or Apache Kafka. [Online] Available: https://content.pivotal.io/blog/understanding-when-to-use-rabbitmq-or-apache-kafka.

[124] ZeroMQ, "Zeromq official documentation," 2020, https://zeromq.org/, Last accessed on 2020-6-16.

[125] T. Taleb, A. Ksentini, and R. Jantti, ""Anything as a Service" for 5G Mobile Systems," *IEEE Network*, vol. 30, no. 6, pp. 84–91, November 2016, doi: http://dx.doi.org/10.1109/MNET.2016.1500244RP.

[126] K. Hwang and D. Suh, "Reducing perceptible IPTV zapping delay using CDN cache server," pp. 738–739, Oct 2013, doi: http://dx.doi.org/10.1109/ICTC.2013.6675467.

[127] A. S. Asrese, S. J. Eravuchira, V. Bajpai, P. Sarolahti, and J. Ott, "Measuring Web Latency and Rendering Performance: Method, Tools, and Longitudinal Dataset," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 535–549, June 2019, doi: http://dx.doi.org/10.1109/TNSM.2019.2896710.

[128] R. Ju, W. Wang, J. Li, F. Li, and L. Han, "On Building a Low Latency Network for Future Internet Services," pp. 1–6, Dec 2017, doi: http://dx.doi.org/10.1109/GLOCOM.2017.8254436.

[129] IDLab. The Virtual Wall. [Online] Available: http://idlab.technology/infrastructure/virtual-wall/.

[130] FED4FIRE Federation for Fire Plus, "Fed4fire+ official documentation," 2020, https://www.fed4fire.eu/, Last accessed on 2020-6-16.

[131] jFed, "jfed official documentation," 2019, https://jfed.ilabt.imec.be/, Last accessed on 2020-6-16.

[132] Docker, "Docker official documentation," 2020, https://docs.docker.com/, Last accessed on 2020-6-16.

[133] Open Baton, "Open baton official documentation," 2017, https://openbaton.github.io/documentation/, Last accessed on 2020-6-16.

[134] Open Source MANO, "Open Source MANO official documentation," 2020, https://osm.etsi.org/, Last accessed on 2020-6-16.

[135] Kubernetes, "Kubernetes official documentation," 2020, https://kubernetes.io/, Last accessed on 2020-6-16.

[136] A. D. Oliva, X. Li, X. Costa-pérez, C. J. Bernardos, P. Bertin, P. Iovanna, T. Deiss, J. Mangues, A. Mourad, C. Casetti, J. E. Gonzalez, and A. Azcorra, "Slicing and Orchestrating Transport Networks for Industry Verticals," no. August, pp. 78–84, 2018, doi: http://dx.doi.org/10.1109/MCOM.2018.1700990.

[137] J. Baranda Hortiguela, J. Mangues-Bafalluy, R. Martinez, L. Vettori, K. Antevski, C. J. Bernardos, and X. Li, "Realizing the Network Service Federation Vision: Enabling Automated Multi-domain Orchestration of Network Services," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 48–57, 2020, doi: http://dx.doi.org/10.1109/MVT.2020.2979558.

[138] 3GPP, "Technical Specification Group Services and System Aspects; Procedures for the 5G System (5GS) Stage 2," *3GPP TS 23.502 V16.6.0*, 2020, online [Available]: https://www.3gpp.org/ftp//Specs/archive/23_series/23.502/.

[139] F. Z. Yousaf, V. Sciancalepore, M. Liebsch, and X. Costa-Perez, "MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 103–109, 2019, doi: https://doi.org/10.1109/MCOM.2019.1800898.

[140] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration," *ETSI ISG NFV, ETSI GS NFV-MAN 001, V1.1.1*, 2014, online [Available]:https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.

[141] 3GPP, "Technical Specification Group Services and System Aspects; System Architecture for the 5G System (5GS) Stage 2," *3GPP TS 23.501 V16.6.0*, 2020, online [Available]: https://www.3gpp.org/ftp//Specs/archive/23_series/23.501/.

[142] 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.

[143] ——, "Deliverable 4.2 - Advanced prototype for secure, cross-border, and multi-domain service orchestration," *H2020 5G-CARMEN Project Consortium*, 2021, online [Available]: https://5gcarmen.eu/publications/.

[144] ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Study on URLLC use cases of vertical industries for DECT evolution and DECT-2020," *ETSI TR 103 515*, 2018, online [Available]: https://www.etsi.org/deliver/etsi_tr/103500_103599/103515/01.01.01_60/tr_103515v010101p.pdf.

[145] S. Maheshwari, D. Raychaudhuri, I. Seskar, and F. Bronzino, "Scalability and Performance Evaluation of Edge Cloud Systems for Latency Constrained Applications," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 286–299, doi: http://dx.doi.org/10.1109/SEC.2018.00028.

[146] 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.

[147] L. Nussbaum, "An overview of Fed4FIRE testbeds – and beyond?" in *GEFI - Global Experimentation for Future Internet Workshop, Coimbra, Portugal*, 2019, online [Available]: https://hal.inria.fr/hal-02401738/document.

[148] J. Struye, B. Braem, S. Latré, and J. Marquez-Barja, "The CityLab testbed — Large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a smart city," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 529–534, doi: http://dx.doi.org/10.1109/INFCOMW.2018.8407018.

[149] M. A. Inamdar and H. V. Kumaraswamy, "Energy Efficient 5G Networks: Techniques and Challenges," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 1317–1322, doi: http://dx.doi.org/10.1109/ICOSEC49089.2020.9215362.

[150] European Commission, "Energy Efficient Cloud Computing Technologies and Policies for an Eco-friendly Cloud Market - Final study report," https://www.francedatacenter.com/wp-content/uploads/2020/11/FINALSTUDYEnglishKK-03-20-210-EN-N13072020pdf.pdf, accessed: 2021-June-22.

[151] T. Norp, "5G Requirements and Key Performance Indicators," *Journal of ICT Standardization*, vol. 6, 2018, doi: https://doi.org/10.13052/jicts2245-800X.612.

[152] J. M. Marquez-Barja, S. Hadiwardoyo, B. Lannoo, W. Vandenberghe, E. Kenis, L. Deckers, M. C. Campodonico, K. dos Santos, R. Kusumakar, M. Klepper, and J. Vandenbossche, "Enhanced Teleoperated Transport and Logistics: A 5G Cross-Border Use Case," in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 229–234, doi: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482459.

[153] J. Marquez-Barja, D. Naudts, V. Maglogiannis, S. A. Hadiwardoyo, I. Moerman, M. Klepper, G. Kakes, L. Xiangyu, W. Vandenberghe, R. Kusumakar, and J. Vandenbossche, "Designing a 5G architecture to overcome the challenges of the teleoperated transport and logistics," *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). pp 1-4. January, 2022. Las Vegas, United States of America.*, 2022, online [Available]: https://www.marquez-barja.com/en/publications.

[154] K. Trichias, G. Landi, E. Seder, J. Marquez-Barja, R. Frizzell, M. Iordache, and P. Demestichas, "VITAL-5G: Innovative Network Applications (NetApps) Support over 5G Connectivity for the Transport & Logistics Vertical," in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 437–442, doi: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482437.

[155] ETSI, "VNF package and PNFD archive specification," *ETSI ISG NFV, ETSI GS NFV-SOL 004 v3.5.1*, 2021, online [Available]: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/004/03.05.01_60/gs_NFV-SOL004v030501p.pdf.

[156] VITAL-5G, "Initial NetApps blueprints and Open Repository design," *VITAL 5G repository*, 2021, online [Available]: https://www.vital5g.eu/wp-content/uploads/2022/01/VITAL5G_D2.1_Initial_NetApps_blueprints_and_Open_Repository_design_Final.pdf.

[157] 3GPP, "Management and orchestration; 5G network resource model (NRM); stage 2 and stage 3," *3GPP TS 28.541 v17.5.0*, 2021, online [Available]: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3400.

[158] T. Doukoglou, V. Gezerlis, K. Trichias, N. Kostopoulos, N. Vrakas, M. Bougioukos, and R. Legouable, "Vertical Industries Requirements Analysis & Targeted KPIs for Advanced 5G Trials," in *2019 European Conference on Networks and Communications (EuCNC)*, 2019, pp. 95–100, doi: https://doi.org/10.1109/EuCNC.2019.8801959.

[159] J. Arjona Aroca, J. A. Giménez Maldonado, G. Ferrús Clari, N. Alonso i García, L. Calabria, and J. Lara, "Enabling a green just-in-time navigation through stakeholder collaboration," *European Transport Research Review*, vol. 12, 2020, doi: https://doi.org/10.1186/s12544-020-00417-7.

[160] M. Oliskevych, S. Kovelyshyn, M. Magats, V. Shevchuk, and O. Sukach, "The optimization of trucks fleet schedule in view of their interaction and restrictions of the european agreement of work of crews," *Transport Problems*, vol. 15, pp. 157–170, 06 2020, doi: https://doi.org/10.21307/tp-2020-028.

[161] R. S. Thomä, C. Andrich, G. Del Galdo, M. Döbereiner, M. A. Hein, M. Käske, G. Schäfer, S. Schieler, C. Schneider, A. Schwind, and P. Wendland, "Cooperative Passive Coherent Location: A Promising 5G Service to Support Road Safety," *IEEE Communications Magazine, note=doi: https://doi.org/10.21307/tp-2020-028*, vol. 57, pp. 86–92, 09 2019.

[162] Y. Ding, M. Jin, S. Li, and D. Feng, "Smart logistics based on the internet of things technology: an overview," *International Journal of Logistics Research and Applications*, vol. 24, no. 4, pp. 323–345, 2021, doi: https://doi.org/10.1080/13675567.2020.1757053.

[163] 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.

[164] 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.

[165] 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.

[166] 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.

[167] 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.

[168] 5GAA, "Use Case Implementation Description," *5GAA Automotive Association*, 2021, online [Available]: https://5gaa.org/wp-content/uploads/2021/04/use-case-t21001.pdf.

[169] 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.

[170] J. Marquez-Barja, B. Lannoo, D. Naudts, B. Braem, C. Donato, V. Maglogiannis, S. Mercelis, R. Berkvens, P. Hellinckx, M. Weyn, I. Moerman, and S. Latre, "Smart Highway: ITS-G5 and C-V2X Based Testbed for Vehicular Communications in Real Environments Enhanced by Edge/Cloud Technologies," *28th European Conference on Networks and Communications (EuCNC), Valencia, Spain*, pp. 1–2, 2019, online [Available]: http://www.marquez-barja.com/images/papers/eucnc-SmartHighway.pdf.

[171] 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)*, 2016, pp. 1–5, doi: https://doi.org/10.1109/VTCFall.2016.7881581.

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

[173] B. Yang, X. Cao, K. Xiong, C. Yuen, Y. L. Guan, S. Leng, L. Qian, and Z. Han, "Edge Intelligence for Autonomous Driving in 6G Wireless System: Design Challenges and Solutions," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 40–47, 2021, doi:https://dx.doi.org/10.1109/MWC.001.2000292.

[174] D. M. Manias and A. Shami, "The Need for Advanced Intelligence in NFV Management and Orchestration," *Netwrk. Mag. of Global Internetwkg.*, vol. 35, no. 1, p. 365–371, Mar. 2021, doi: https://dx.doi.org/10.1109/MNET.011.2000373. [Online]. Available: https://doi.org/10.1109/MNET.011.2000373

[175] M. Camelo, L. Cominardi, M. Gramaglia, M. Fiore, A. Garcia-Saavedra, L. Fuentes, D. D. Vleeschauwer, P. Soto, N. Slamnik-Kriještorac, J. Ballesteros, C.-Y. Chang, G. Baldoni, J. M. Marquez-Barja, P. Hellinckx, and S. Latré,

"Requirements and Specifications for the Orchestration of Network Intelligence in 6G," Dec. 2021, doi:https://dx.doi.org/10.5281/zenodo.5767861. [Online]. Available: https://doi.org/10.5281/zenodo.5767861

[176] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021, doi:https://dx.doi.org/10.1016/j.neucom.2021.07.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221010894

[177] ETSI, "Autonomous Networks, supporting tomorrow's ICT business," *ETSI White Paper No. 40*, no. 40, 2020, online [Available]: https://www.etsi.org/images/files/ETSIWhitePapers/etsi-wp-40-Autonomous-networks.pdf.

[178] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020, doi: https://dx.doi.org/10.23919/JCC.2020.09.009.

[179] S. Disabato and M. Roveri, "Incremental On-Device Tiny Machine Learning," in *Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, ser. AIChallengeIoT '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 7–13, doi:https://dx.doi.org/10.1145/3417313.3429378. [Online]. Available: https://doi.org/10.1145/3417313.3429378

[180] G. Baldoni, J. Loudet, L. Cominardi, A. Corsaro, and Y. He, "Facilitating Distributed Data-Flow Programming with Eclipse Zenoh: The ERDOS Case," in *Proceedings of the 1st Workshop on Serverless Mobile Networking for 6G Communications*, ser. MobileServerless'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 13–18, doi:https://dx.doi.org/10.1145/3469263.3469858. [Online]. Available: https://doi.org/10.1145/3469263.3469858

[181] 3GPP, "Application layer support for Vehicle-to-Everything (V2X) services; Functional architecture and information flows , year=2019, note=Online [Available]: https://www.3gpp.org/ftp/Specs/archive/23_series/23.286/,," *3GPP Technical specification (TS)*.

[182] G. Shuxin, M. Cheng, X. He, and X. Zhou, "A Two-Stage Service Migration Algorithm in Parked Vehicle Edge Computing for Internet of Things," *Sensors*, vol. 20, no. 10, 2020, doi: https://doi.org/10.3390/s20102786. [Online]. Available: https://www.mdpi.com/1424-8220/20/10/2786

[183] L. Pacheco, D. Rosário, E. Cerqueira, L. Villas, T. Braun, and A. A. F. Loureiro, "Distributed user-centric service migration for edge-enabled networks," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 618–622, online [Available]: https://ieeexplore.ieee.org/document/9463944.

[184] P. Soto, D. De Vleeschauwer, M. Camelo, Y. De Bock, K. De Schepper, C.-Y. Chang, P. Hellinckx, J. F. Botero, and S. Latré, "Towards Autonomous VNF Auto-scaling using Deep Reinforcement Learning," *Zenodo*, Dec. 2021, doi: https://doi.org/10.5281/zenodo.5767618. [Online]. Available: https://doi.org/10.5281/zenodo.5767618

[185] J. Marquez-Barja, B. Lannoo, D. Naudts, B. Braem, V. Maglogiannis, C. Donato, S. Mercelis, R. Berkvens, P. Hellinckx, M. Weyn *et al.*, "Smart Highway: ITS-G5 and C2VX based testbed for vehicular communications in real environments enhanced by edge/cloud technologies," in *EuCNC2019, the European Conference on Networks and Communications*, 2019, pp. 1–2, available [Online]:https://biblio.ugent.be/publication/8642435.

[186] J. Baranda and et al., "On the Integration of AI/ML-based scaling operations in the 5Growth platform," in *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2020, pp. 105–109, doi: https://dx.doi.org/10.1109/NFV-SDN50289.2020.9289863.

[187] N. Slamnik-Krieštorac and J. M. Marquez-Barja, "Unraveling Edge-based in-vehicle infotainment using the Smart Highway testbed," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–4, doi: https://dx.doi.org/10.1109/CCNC49032.2021.9369622.

[188] N. Slamnik-Krieštorac, G. M. Yilma, F. Zarrar Yousaf, M. Liebsch, and J. M. Marquez-Barja, "Multi-domain MEC orchestration platform for enhanced Back Situation Awareness," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–2, doi: https://dx.doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484632.

[189] Eclipse zenoh, "Project website: Eclipse zenoh," 2020, online [Available]: https://zenoh.io, Last accessed on 2021-9-20.

[190] J. Zhang and K. B. Letaief, "Mobile Edge Intelligence and Computing for the Internet of Vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2020, doi: https://doi.org/10.1109/JPROC.2019.2947490.

[191] A. Kanavos, D. Fragkos, and A. Kaloxylos, "V2X Communication over Cellular Networks: Capabilities and Challenges," *Telecom*, vol. 2, no. 1, pp. 1–26, 2021, doi: http://dx.doi.org/10.3390/telecom2010001. [Online]. Available: https://www.mdpi.com/2673-4001/2/1/1

[192] 3GPP, "3GPP Architecture for Enabling Edge Applications," https://www.3gpp.org/ftp/Specs/archive/23_series/23.558/, accessed: 2021-04-12.

[193] J. Violos, E. Psomakelis, D. Danopoulos, S. Tsanakas, and T. Varvarigou, *Using LSTM Neural Networks as Resource Utilization Predictors: The Case of Training Deep Learning Models on the Edge*, 12 2020, pp. 67–74, doi: http://dx.doi.org/10.1007/978-3-030-63058-4_6.

[194] A. Singh, "Major MCDM Techniques and their application-A Review," *IOSR Journal of Engineering*, vol. 4, pp. 15–25, 05 2014, doi: http://dx.doi.org/10.9790/3021-04521525.

[195] M. Sabaghi, C. Mascle, and P. Baptiste, "Application of DOE-TOPSIS Technique in Decision-Making Problems," *15th IFAC Symposium on Information Control Problems in Manufacturing*, vol. 48, no. 3, pp. 773–777, 2015, doi: https://dx.doi.org/10.1016/j.ifacol.2015.06.176. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315004152

[196] M. Awad and R. Khanna, "Support Vector Regression. In: Efficient Learning Machines." *Apress, Berkeley, CA*, 2015, doi: https://doi.org/10.1007/978-1-4302-5990-9_4.

[197] 5GAA, "Safety Treatment in Connected and Autonomous Driving Functions Report," *5GAA Automotive Association*, 2021, online [Available]: https://5gaa.org/wp-content/uploads/2021/04/use-case-t21001.pdf.

[198] N. Slamnik-Kriještorac, S. Latré, and J. M. Marquez-Barja, "An optimized application-context relocation approach for Connected and Automated Mobility (CAM)," in *IEEE 5G for CAM - 5G Virtual Summit*, 2021, online [Available]: https://arxiv.org/abs/2109.11362.

[199] S. Wang, T. Sun, H. Yang, X. Duan, and L. Lu, "6G Network: Towards a Distributed and Autonomous System," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5, doi: https://doi.org/10.1109/6GSUMMIT49458.2020.9083888.

[200] D. Appleby, "Shared goods," pp. 128–150, [Online] Available: https://bit.ly/2Uxy6AN.

[201] Y. Al-Yasir, *Fundamentals of 5G mobile networks*, 05 2018, [Online] Available: https://www.researchgate.net/publication/324862482_Fundamentals_of_5G_mobile_networks.

[202] B. Martucci. (2018, october) What is the sharing economy – example companies, definition, pros & cons. [Online] Available: https://bit.ly/2k81Vm4.

[203] K. McBride, "Sharing cities: a case for truly smart and sustainable cities by Duncan McLaren and Julian Agyeman, Cambridge, MA, MIT Press, 2015, 445 pp., $32.00 (hardback), 978-0-262-02972-8 ," *Urban Geography*, 2016, doi: http://dx.doi.org/10.1080/02723638.2016.1235934.

[204] V. Behrends, J. G. Bundy, A. B. Phillimore, T. Bell, D. Lawrence, T. G. Barraclough, and F. Fiegna, "Species Interactions Alter Evolutionary Responses to a Novel Environment," *PLoS Biology*, vol. 10, no. 5, p. e1001330, 2012, doi: http://dx.doi.org/10.1371/journal.pbio.1001330.

[205] J. G. Ojalvo, "Dynamical strategies for resource sharing in bacteria Dynamical strategies for resource sharing in bacteria," 2018, [Online] Available: http://www.solvayinstitutes.be/event/workshop/dynamics_2018/talks/garcia_ojalvo.pdf.

[206] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Game-Theoretic infrastructure sharing in multioperator cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3326–3341, 2016, doi: http://dx.doi.org/10.1109/TVT.2015.2445837.

[207] N. Afraz and M. Ruffini, "A Sharing Platform for Multi-Tenant PONs," *Journal of Lightwave Technology*, vol. 36, no. 23, pp. 5413–5423, Dec 2018, doi: http://dx.doi.org/10.1109/JLT.2018.2875188.

[208] J. M. Márquez-Barja, M. Ruffini, N. J. Kaminski, N. Marchetti, L. Doyle, and L. A. DaSilva, "Decoupling Resource Ownership From Service Provisioning to Enable Ephemeral Converged Networks (ECNs)," 2016, [Online] Available: https://bit.ly/2InwwtL.

[209] Roya H.Tehrani, V. Seiamak, T. Dionysia, L. Haeyoung, and M. Klaus, "Licensed Spectrum Sharing Schemes for Mobile Operators : A Survey and Outlook," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 1–33, 2016, doi: http://dx.doi.org/10.1109/COMST.2016.2583499.

[210] G. R. Nair, Y. K. Moorthy, and S. S. Pillai, "A Survey on Dynamic Spectrum Sharing Using Game Theory In Cognitive Radio Networks," *International Journal of Research and Engineering*, vol. 03, no. 08, 2016, [Online] Available: https://core.ac.uk/download/pdf/154060409.pdf.

[211] F. Hu, B. Chen, and K. Zhu, "Full Spectrum Sharing in Cognitive Radio Networks Toward 5G: A Survey," *IEEE Access*, vol. 6, no. c, pp. 15 754–15 776, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2802450.

[212] L. Zhang, Y. C. Liang, and M. Xiao, " Sharing for Internet of Things: A Survey," *IEEE Wireless Communications*, 2018, doi: http://dx.doi.org/10.1109/MWC.2018.1800259.

[213] A. M. Voicu, L. Simic, and M. Petrova, "Survey of Spectrum Sharing for Inter-Technology Coexistence," *IEEE Communications Surveys and Tutorials*, no. i, pp. 1–33, 2018, doi: http://dx.doi.org/10.1109/COMST.2018.2882308.

[214] L. Zhang, M. Xiao, G. Wu, M. Alam, Y. C. Liang, and S. Li, "A Survey of Advanced Techniques for Spectrum Sharing in 5G Networks," pp. 44–51, 2017, doi: http://dx.doi.org/10.1109/MWC.2017.1700069.

[215] Y. Ye, D. Wu, Z. Shu, and Y. Qian, "Overview of LTE Spectrum Sharing Technologies," *IEEE Access*, vol. 4, no. c, pp. 8105–8115, 2016, doi: http://dx.doi.org/10.1109/ACCESS.2016.2626719.

[216] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, and G. N. Rouskas, "Spectrum management techniques for elastic optical networks: A survey," *Optical Switching and Networking*, vol. 13, no. 2, pp. 34–48, 2014, doi: http://dx.doi.org/10.1016/j.osn.2014.02.003. [Online]. Available: http://dx.doi.org/10.1016/j.osn.2014.02.003

[217] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, 2009, doi: http://dx.doi.org/10.1109/JPROC.2009.2015717.

[218] E. P. Goodman, "Spectrum Rights in the Telecosm to Come," *San Diego Law Review*, vol. 41, p. 269, 2004, [Online] Available: https://bit.ly/2Ho3LMu. [Online]. Available: http://heinonline.org.ezproxy.library.wisc.edu/HOL/Page?handle=hein.journals/sanlr41{&}id=281{&}div={&}collection=journals{%}5Cnhttp://heinonline.org.ezproxy.library.wisc.edu/HOL/Page?handle=hein.journals/sanlr41{&}div=22{&}collection=journals{&}set{_}as{_}cursor=29{&}men{_}ta

[219] T. W. Hazlett and B. Skourp, "Tragedy of the Regulatory Commons: Lightsquared and Missing Spectrum Rights," *Duke Law & Technology Review*, vol. 13, no. January, 2014, [Online] Available: https://bit.ly/2YypuY0. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=2544673

[220] W. S. H. M. W. Ahmad, N. A. M. Radzi, and F. S. Samidi, "5G technology: Towards Dynamic Spectrum Sharing using Cognitive Radio Networks, volume = PP, year = 2020," *IEEE Access*, p. 1, doi: http://dx.doi.org/10.1109/ACCESS.2020.2966271.

[221] S. Zahoor and R. N. Mir, "Virtualization and iot resource management: A survey," *International Journal of Computer Networks And Applications*, vol. 5, no. 4, p. 43, 2018, doi: http://dx.doi.org/10.22247/ijcna/2018/49435.

[222] A. P. Bianzino, C. Chaudet, D. Rossi, and J. L. Rougier, "A survey of green networking research," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012, doi: http://dx.doi.org/10.1109/SURV.2011.113010.00106.

[223] L. Mamushiane, A. A. Lysko, and S. Dlamini, "SDN-enabled Infrastructure Sharing in Emerging Markets: CapEx/OpEx Savings Overview and Quantification," *2018 IST-Africa Week Conference (IST-Africa)*, pp. Page 1 of 10–Page 10 of 10, 2018, [Online] Available: https://bit.ly/2D9E2oD.

[224] M. Ali, "Shareability in optical networks: beyond bandwidth optimization," *IEEE Communications Magazine*, vol. 42, no. 2, pp. S11–S15, feb 2004, doi: http://dx.doi.org/10.1109/MCOM.2003.1267096. [Online]. Available: http://ieeexplore.ieee.org/document/1267096/

[225] O. Gerstel, I. , D. Klonidis, I. Tomkos, and E. Palkopoulou, "Dynamic Cooperative Spectrum Sharing and Defragmentation for Elastic Optical Networks," *Journal of Optical Communications and Networking*, vol. 6, no. 3, p. 259, 2014, doi: http://dx.doi.org/10.1364/jocn.6.000259.

[226] N. Afraz, A. Elrasad, and M. Ruffini, "DBA Capacity Auctions to Enhance Resource Sharing across Virtual Network Operators in Multi-Tenant PONs," pp. 1–3, March 2018, [Online] Available: https://ieeexplore.ieee.org/document/8386208.

[227] Y. Shi, M. Sheng, and F. He, "A resource management and control model supporting applications in the internet of things," 2011, doi: http://dx.doi.org/10.1109/iThings/CPSCom.2011.27.

[228] A. M. Akhtar, X. Wang, and L. Hanzo, "Synergistic spectrum sharing in 5G HetNets: A harmonized SDN-enabled approach," *IEEE Communications Magazine*, 2016, doi: http://dx.doi.org/10.1109/MCOM.2016.7378424.

[229] H. Kour, R. K. Jha, and S. Jain, "A comprehensive survey on spectrum sharing: Architecture, energy efficiency and security issues," *Journal of Network and Computer Applications*, vol. 103, no. November, pp. 29–57, 2018, doi: http://dx.doi.org/10.1016/j.jnca.2017.11.010.

[230] Y. Han, E. Ekici, H. Kremo, and O. Altintas, "Spectrum sharing methods for the coexistence of multiple RF systems: A survey," *Ad Hoc Networks*, vol. 53, pp. 53–78, 2016, doi: http://dx.doi.org/10.1016/j.adhoc.2016.09.009.

[231] R. Gould and J. Kelleher, "Frequency Sharing Between the Broadcasting-Satellite Service and Other Radiocommunication Services," *IEEE Journal on Selected Areas in Communications*, vol. 3, no. 1, pp. 25–35, 2008, doi: http://dx.doi.org/10.1109/jsac.1985.1146190.

[232] T. A. Prosch, "A possible frequency planning method and related model calculations for the sharing of VHF band ii (87.5-108 MHz) between fm and dab (digital audio broadcast) systems," pp. 55–63, 1991, doi: http://dx.doi.org/10.1109/11.86962.

[233] V. K. Varma, H. W. Arnold, D. M. J. Devasirvatham, A. Ranade, and L. G. Sutliff, "Interference, Sensitivity and capacity Analysis for Measurement-Based Wireless Access Spectrum Sharing," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 611–616, 1994, doi: http://dx.doi.org/10.1109/25.312812.

[234] S. Tridandapani and B. Mukherjee, "Multicast traffic in multi-hop lightwave networks: performance analysis and an argument for channel sharing," vol. 15, no. 3, pp. 488–500, 1997, doi: http://dx.doi.org/10.1109/49.564144.

[235] G. J. Foschini, "Sharing of the Optical Band in Local Systems," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 6, pp. 974–986, 1988, doi: http://dx.doi.org/10.1109/49.1960.

[236] P. Papadimitratos, S. Sankaranarayanan, and A. Mishra, "A Bandwidth Sharing Approach to Improve Licensed Spectrum Utilization," *IEEE Communications Magazine*, vol. 43, no. 12, pp. S10–S14, 2005, doi: http://dx.doi.org/10.1109/MCOM.2005.1561918.

[237] J. M. Peha, "Approaches to spectrum sharing," *IEEE Communications Magazine*, vol. 43, no. 2, 2005, doi: http://dx.doi.org/10.1109/MCOM.2005.1391490.

[238] J. Hultell, K. Johansson, and J. Markendahl, "Business models and resource management for shared wireless networks," no. May, pp. 3393–3397, 2005, doi: http://dx.doi.org/10.1109/vetecf.2004.1404693.

[239] F. C. Commission, "Report and Order and Order on Demand and Further Notice of Proposed Rulemaking," *FCC's proceedings*, 2003, [Online] Available: https://bit.ly/2Kbmcqn. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=2544673

[240] W. Jones, "Share and share not," *IEEE Spectrum*, vol. 40, no. 4, pp. 19–21, 2003, doi: http://dx.doi.org/10.1109/mspec.2003.1191779.

[241] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, 2006, doi: http://dx.doi.org/10.1016/j.comnet.2006.05.001.

[242] H. Sarvanko, M. Höyhtyä, M. Katz, and F. H. P. Fitzek, "Distributed resources in wireless networks: Discovery and cooperative uses," *ERCIM eMobility Workshop*, 2010, doi: http://dx.doi.org/10.1073/pnas.0703993104. [Online]. Available: http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:Distributed+Resources+in+Wireless+Networks+:+Discovery+and+Cooperative+Uses{#}0

[243] I. F. Akyildiz, W.-y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40–48, 2008, doi: http://dx.doi.org/10.1109/MCOM.2008.4481339.

[244] J. M. Peha, "Sharing Spectrum Through Spectrum Policy Reform and Cognitive Radio ," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 708–719, 2009, doi: http://dx.doi.org/10.1109/JPROC.2009.2013033.

[245] C. M. Sudharman K. Jayaweera, Gonzalo Vazquez-Vilar, "Dynamic Spectrum Leasing: A New Paradigm for Spectrum Sharing in Cognitive Radio Networks ," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2328–2339, 2010, doi: http://dx.doi.org/10.1109/TVT.2010.2042741.

[246] P. Si, H. Ji, F. R. Yu, and V. C. M. Leung, "Optimal cooperative internetwork spectrum sharing for cognitive radio systems with spectrum pooling," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1760–1768, 2010, doi: http://dx.doi.org/10.1109/TVT.2010.2041941.

[247] K. B. L. Karama Hamdi, Wei Zhang, "Opportunistic spectrum sharing in cognitive MIMO wireless networks ," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 4098–4109, 2009, doi: http://dx.doi.org/10.1109/TWC.2009.080528.

[248] M. J. Marcus, "Sharing government spectrum with private users: opportunities and challenges ," *IEEE Wireless Communications*, vol. 16, no. 3, pp. 4–5, 2009, doi: http://dx.doi.org/10.1109/MWC.2009.5109457.

[249] D. E. Meddour, T. Rasheed, and Y. Gourhant, "On the role of infrastructure sharing for mobile network operators in emerging markets," *Computer Networks*, 2011, doi: http://dx.doi.org/10.1016/j.comnet.2011.01.023.

[250] P. J. del Cid, S. Michiels, W. Joosen, and D. Hughes, "Middleware for resource sharing in multi-purpose wireless sensor networks," pp. 1–8, 2010, doi: http://dx.doi.org/10.1109/NESEA.2010.5678061.

[251] T. E. Darcie, N. Barakat, P. P. Iannone, and K. C. Reichmann, "Wavelength sharing in WDM passive optical networks," *Optical Transmission, Switching, and Subsystems VI*, vol. 7136, p. 71361I, 2008, doi: http://dx.doi.org/10.1117/12.806535.

[252] J. Kibilda and L. A. Dasilva, "Efficient coverage through inter-operator infrastructure sharing in mobile networks," 2013, doi: http://dx.doi.org/10.1109/WD.2013.6686480.

[253] E. Jorswieck, L. Badia, T. Fahldieck, E. Karipidis, and J. Luo, "Spectrum sharing improves the network efficiency for cellular operators," *IEEE Communications Magazine*, 2014, doi: http://dx.doi.org/10.1109/MCOM.2014.6766097.

[254] J. M. Park, J. H. Reed, A. A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceedings of the IEEE*, 2014, doi: http://dx.doi.org/10.1109/JPROC.2014.2301972.

[255] TCCA. (2014, March) Protected spectrum critical for the future of public safety. [Online] Available: https://tcca.info/protected-spectrum-critical-for-the-future-of-public-safety/.

[256] M. De Leenheer, J. Buysse, C. Develder, and B. Mukherjee, "Isolation and resource efficiency of virtual optical networks," 2012, doi: http://dx.doi.org/10.1109/ICCNC.2012.6167543.

[257] R. Vilalta, R. Muñoz, R. Casellas, and R. Martinez, "Dynamic virtual GMPLS-controlled WSON using a Resource Broker with a VNT Manager on the ADRENALINE testbed," *Optics Express*, 2012, doi: http://dx.doi.org/10.1364/oe.20.029149.

[258] F. A. Khandaker, J. P. Jue, X. Wang, Q. Zhang, Q. She, H. C. Cankaya, P. Palacharla, and M. Sekiya, "Statistical capacity sharing for variable-rate connections in flexible grid optical networks," 2015, doi: http://dx.doi.org/10.1109/GLOCOM.2014.7417756.

[259] F. A. Khandaker, X. Wang, Q. Zhang, H. C. Cankaya, I. Kim, T. Ikeuchi, and J. P. Jue, "Statistical sharing of primary and back-up capacity in survivable elastic optical networks," pp. 1–6, 2017, doi: 10.1109/GLOCOM.2017.8254765.

[260] X. Wang, Q. Zhang, I. Kim, P. Palacharla, and M. Sekiya, "Support Statistical Sharing in Circuit Switching WDM Optical Networks," 2013, doi: http://dx.doi.org/10.1364/ofc.2013.otu3a.1.

[261] O. Pedrola, D. Careglio, M. Klinkowski, J. Solé-Pareta, and K. Bergman, "Cost Feasibility Analysis of Translucent Optical Networks With Shared Wavelength Converters," *Journal of Optical Communications and Networking*, 2013, doi: http://dx.doi.org/10.1364/jocn.5.000104.

[262] A. Manolova, I. Cerutti, R. Muñoz, S. Ruepp, A. Giorgetti, N. Andriolli, N. Sambo, P. Castoldi, R. Martínez, and R. Casellas, "Distributed Sharing of Functionalities and Resources in Survivable GMPLS-Controlled WSONs," *Journal of Optical Communications and Networking*, 2012, doi: http://dx.doi.org/10.1364/jocn.4.000219.

[263] E. Palkopoulou, I. Stiakogiannakis, D. Klonidis, K. Christodoulopoulos, E. Varvarigos, O. Gerstel, and I. Tomkos, "Dynamic Cooperative Spectrum Sharing in Elastic Networks," 2013, doi: http://dx.doi.org/110.1364/ofc.2013.otu3a.2.

[264] R. Silva, J. Sa Silva, and F. Boavida, "A symbiotic resources sharing IoT platform in the smart cities context," *2015 IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2015*, no. April, pp. 7–9, 2015, doi: http://dx.doi.org/10.1109/ISSNIP.2015.7106922.

[265] A. Kliem and O. Kao, "The Internet of Things Resource Management Challenge," *Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE International Conference on Internet of Things, DSDIS/CPSCom/GreenCom/iThings 2015*, pp. 483–490, 2015, doi: http://dx.doi.org/10.1109/DSDIS.2015.21.

[266] M. Rebato, M. Mezzavilla, S. Rangan, and M. Zorzi, "The Potential of Resource Sharing in 5G Millimeter-Wave Bands," no. 14, 2016, [Online] Available: https://bit.ly/1S4Nr2N.

[267] A. K. Gupta, J. G. Andrews, and R. W. Heath, "On the Feasibility of Sharing Spectrum Licenses in mmWave Cellular Systems," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3981–3995, 2016, doi: http://dx.doi.org/10.1109/TCOMM.2016.2590467.

[268] L. Wan and H. T. Co, "4G/5G Spectrum Sharing for Enhanced Mobile Broad-Band and IoT Services," *IEEE Vehicular Technology Magazine*, p. 10, 2018, doi: http://dx.doi.org/10.1109/MVT.2018.2865830.

[269] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV : A survey of taxonomy , architectures and future challenges," vol. 167, 2020, doi: http://dx.doi.org/10.1016/j.comnet.2019.106984.

[270] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017, doi: http://dx.doi.org/10.1109/MCOM.2017.1600940.

[271] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network Slicing for 5G: Challenges and Opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017, doi: http://dx.doi.org/10.1109/MIC.2017.3481355.

[272] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, G. Fu, and Z. Zhu, "Application-Driven End-to-End Slicing: When Wireless Network Virtualization Orchestrates With NFV-Based Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 26 567–26 577, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2834623.

[273] D. B. Rawat, "Fusion of Software Defined Networking , Edge Computing , and Blockchain Technology for Wireless Network Virtualization," *IEEE Communications Magazine*, vol. 57, no. October, pp. 50–55, 2019, doi: http://dx.doi.org/10.1109/MCOM.001.1900196.

[274] B. Yin, W. Shen, Y. Cheng, L. X. Cai, and Q. Li, "Distributed Resource Sharing in Fog-assisted Big Data Streaming," pp. 1–6, May 2017, doi: http://dx.doi.org/10.1109/ICC.2017.7996724.

[275] A. Elrasad, N. Afraz, and M. Ruffini, "Virtual dynamic bandwidth allocation enabling true PON multi-tenancy," pp. 1–3, March 2017, online [Available]: https://ieeexplore.ieee.org/document/7936877.

[276] A. Elrasad and M. Ruffini, "Frame Level Sharing for DBA virtualization in multi-tenant PONs," pp. 1–6, May 2017, doi: http://dx.doi.org/10.23919/ONDM.2017.7958528.

[277] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti, "Demonstration of Dynamic Resource Sharing Benefits in an Optical C-RAN," *Journal of Optical Communications and Networking*, vol. 8, no. 8, p. 621, 2016, doi: http://dx.doi.org/10.1364/jocn.8.000621.

[278] P. M. et. al., "Optical and wireless network convergence in 5G systems - an experimental approach," 2018, doi: http://dx.doi.org/10.1109/CAMAD.2018.8514972.

[279] C. K. Dominicini, C. B. Both, J. A. Wickboldt, R. Nejabati, D. F. Macedo, J. Marquez-Barja, and L. da Silva, "Enabling Experimental Research Through Converged Orchestration of Optical Wireless and Cloud Domains," *European Conference on Networks and Communications (EuCNC)*, no. i, pp. 1–2, 2018, [Online] Available: https://bit.ly/2wLqsUV.

[280] P. Alvarez, F. Slyne, C. Bluemm, J. M. Marquez-Barja, L. A. DaSilva, and M. Ruffini, "Experimental Demonstration of SDN-controlled Variable-rate Fronthaul for Converged LTE-over-PON," 2018, doi: http://dx.doi.org/0.1364/ofc.2018.th2a.49.

[281] F. Slyne, R. Guimaraes, Y. Zhang, M. Martinello, R. Nejabati, M. Ruffini, and L. DaSilva, "Coordinated fibre and wireless spectrum allocation in SDN-controlled wireless-optical-cloud converged architecture," pp. 1–3, September 2019, online [Available]: https://www.ecoc2019.org/demo.html.

[282] E. Municio, J. Marquez-Barja, S. Latré, and S. Vissicchio, "Whisper: Programmable and flexible control on industrial IoT networks," *Sensors (Switzerland)*, 2018, doi: http://dx.doi.org/10.3390/s18114048.

[283] N. Kouvelas, V. Balasubramanian, A. G. Voyiatzis, R. R. Prasad, and D. Pesch, "On inferring how resources are shared in IoT ecosystems; a graph theoretic approach," *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, vol. 2018-January, pp. 760–766, 2018, doi: http://dx.doi.org/10.1109/WF-IoT.2018.8355137.

[284] G. Yildirim and Y. Tatar, "Simplified Agent-Based Resource Sharing Approach for WSN-WSN Interaction in IoT/CPS Projects," *IEEE Access*, vol. 6, no. c, pp. 78 077–78 091, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2884741.

[285] N. S. Vo, T. Q. Duong, M. Guizani, and A. Kortun, "5G optimized caching and downlink resource sharing for smart cities," *IEEE Access*, vol. 6, pp. 31 457–31 468, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2839669.

[286] L. Tom, C. Tselios, D. M. Area, and G. Tsolis, "On the deployment of an open-source, 5G-aware evaluation testbed," 2018, doi: http://dx.doi.org/10.1109/MobileCloud.2018.00016.

[287] A. Kostopoulos, G. Agapiou, F. C. Kuo, K. Pentikousis, A. Cipriano, D. Panaitopol, D. Marandin, K. Kowalik, K. Alexandris, C. Y. Chang, N. Nikaein, M. Goldhamer, A. Kliks, R. Steinert, A. Mammela, and T. Chen, "Scenarios for 5G networks: The COHERENT approach," *2016 23rd International Conference on Telecommunications, ICT 2016*, 2016, doi: http://dx.doi.org/10.1109/ICT.2016.7500421.

[288] J. Isnard, "Frequency band sharing: utopia or reality? Towards specification of operational scenarios," *IEEE Aerospace and Electronic Systems Magazine*, vol. 17, no. 5, pp. 4–9, 2002, doi: http://dx.doi.org/10.1109/62.1001985.

[289] S. Satkunarajah, K. Ratnam, and R. G. Ragel, "Pre-configured backup protection with limited resource sharing in elastic optical networks," *2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015 - Conference Proceedings*, pp. 513–518, 2016, doi: http://dx.doi.org/10.1109/ICIINFS.2015.7399065.

[290] B. Chen, Y. Zhao, and J. Zhang, "Survivable spectrum-shared ability in flexible bandwidth optical networks with distributed data centers," *Photonic Network Communications*, vol. 33, no. 2, pp. 102–111, 2017, doi: http://dx.doi.org/10.1007/s11107-016-0642-3.

[291] S. N. Khan, L. Goratti, R. Riggio, and S. Hasan, "On active, fine-grained RAN and spectrum sharing in multi-tenant 5G networks," 2018, doi: http://dx.doi.org/10.1109/PIMRC.2017.8292672.

[292] F. Fund, S. Shahsavari, S. S. Panwar, E. Erkip, and S. Rangan, "Spectrum and Infrastructure Sharing in Millimeter Wave Cellular Networks: An Economic Perspective," 2016, [Available] Online: https://bit.ly/2LcBpJ5. [Online]. Available: http://arxiv.org/abs/1605.04602

[293] P. Luoto, P. Pirinen, M. Bennis, S. Samarakoon, S. Scott, and M. Latva-Aho, "Co-primary multi-operator resource sharing for small cell networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3120–3130, 2015, doi: http://dx.doi.org/10.1109/TWC.2015.2402671.

[294] M. A. Marotta, N. Kaminski, I. Gomez-Miguelez, L. Z. Granville, J. Rochol, L. DaSilva, and C. B. Both, "Resource sharing in heterogeneous cloud radio access networks," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 74–82, 2015, doi: http://dx.doi.org/10.1109/MWC.2015.7143329.

[295] M. Rebato, M. Mezzavilla, S. Rangan, and M. Zorzi, "Resource sharing in 5G mmWave cellular networks," *Proceedings - IEEE INFOCOM*, vol. 2016-September, no. mmNet, pp. 271–276, 2016, doi: http://dx.doi.org/10.1109/INFCOMW.2016.7562085.

[296] R. Yu, J. Ding, X. Huang, M. T. Zhou, S. Gjessing, and Y. Zhang, "Optimal Resource Sharing in 5G-Enabled Vehicular Networks: A Matrix Game Approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7844–7856, 2016, doi: http://dx.doi.org/10.1109/TVT.2016.2536441.

[297] G. Salami, O. Durowoju, A. Attar, O. Holland, R. Tafazolli, and H. Aghvami, "A comparison between the centralized and distributed approaches for spectrum management," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 2, pp. 274–290, 2011, doi: http://dx.doi.org/10.1109/SURV.2011.041110.00018.

[298] J. M. Batalla, C. X. Mavromoustakis, G. Mastorakis, and K. Sienkiewicz, "Internet of Things (IoT) in 5G Mobile Technologies," vol. 8, pp. 25–36, 2016, doi: http://dx.doi.org/10.1007/978-3-319-30913-2. [Online]. Available: http://link.springer.com/10.1007/978-3-319-30913-2

[299] Ning-Hai Bao, Le-Min Li, Hong-Bin Luo, Zhi-Zhong Zhang, and Hong-Fang Yu, "On Exploiting Sharable Resources With Resource Contention Resolution for Surviving Double-Link Failures in Optical Mesh Networks," *Journal of Lightwave Technology*, vol. 30, no. 17, pp. 2788–2795, 2012, doi: http://dx.doi.org/10.1109/jlt.2012.2208178.

[300] J. Thangaraj, P. D. Mankar, and R. Datta, "Improved shared resource allocation strategy with SLA for survivability in WDM optical networks," *Journal of Optics*, vol. 39, no. 2, pp. 57–75, 2011, doi: http://dx.doi.org/10.1007/s12596-010-0022-9.

[301] C. Li, W. Guo, W. Wang, W. Hu, and M. Xia, "Bandwidth Resource Sharing on the XG-PON Transmission Convergence Layer in a Multi-operator Scenario," *Journal of Optical Communications and Networking*, vol. 8, no. 11, p. 835, 2016, doi: http://dx.doi.org/10.1364/jocn.8.000835.

[302] C. Beckman and G. Smith, "Accepted from open call - Shared networks: making wireless communication affordable," *IEEE Wireless Communications*, vol. 12, no. 2, pp. 78–85, 2005, doi: http://dx.doi.org/10.1109/mwc.2005.1421931.

[303] A. Pagani and K. Mikhaylov, "Resource sharing between neighboring nodes in heterogeneous wireless sensor networks," pp. 522–527, 2015, doi: http://dx.doi.org/10.1109/EuCNC.2015.7194130.

[304] T. Frisanco, P. Tafertshofer, P. Lurin, and R. Ang, "Infrastructure sharing and shared operations for mobile network operators from a deployment and operations view," pp. 129–136, April 2008, doi: http://dx.doi.org/10.1109/NOMS.2008.4575126.

[305] V. Angelakis, I. Avgouleas, N. Pappas, E. Fitzgerald, and D. Yuan, "Allocation of Heterogeneous Resources of an IoT Device to Flexible Services," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 691–700, 2016, doi: http://dx.doi.org/10.1109/JIOT.2016.2535163.

[306] C. Li, W. Guo, W. Wang, W. Hu, and M. Xia, "PON bandwidth resource sharing schemes in a multi-operator scenario," pp. 397–401, 2017, doi: http://dx.doi.org/10.1109/ICCNC.2017.7876161.

[307] M. Ruffini and F. Slyne, "Moving the Network to the Cloud: The Cloud Central Office Revolution and Its Implications for the Optical Layer," *J. Lightwave Technol.*, vol. 37, no. 7, pp. 1706–1716, Apr 2019, online [Available]: http://jlt.osa.org/abstract.cfm?URI=jlt-37-7-1706.

[308] F. Slyne, A. Elrasad, C. Bluemm, and M. Ruffini, "Demonstration of Real Time VNF Implementation of OLT with Virtual DBA for Sliceable Multi-Tenant PONs," pp. 1–3, March 2018, online [Available]: https://ieeexplore.ieee.org/document/8385950.

[309] N. Afraz and M. Ruffini, "A distributed bilateral resource market mechanism for future telecommunications networks," Dec. 2019, online [Available]: https://www.linkedin.com/pulse/distributed-bilateral-resource-market-mechanism-future-nima-afraz/.

[310] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," 2016, [Online] Available: https://dl.acm.org/doi/abs/10.1145/2999572.2999599.

[311] P. Shantharama, A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione, "Layback: Sdn management of multi-access edge computing (mec) for network access services and radio resource sharing," *IEEE Access*, vol. 6, pp. 57 545–57 561, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2873984.

[312] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on ran: Flexibility and resources abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, June 2017, doi: http://dx.doi.org/10.1109/MCOM.2017.1601119.

[313] S. Nadas, Z. Turanyi, G. Gombos, and S. Laki, "Stateless resource sharing in networks with multi-layer virtualization," pp. 1–7, May 2019, doi: http://dx.doi.org/10.1109/ICC.2019.8761720.

[314] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2429–2453, thirdquarter 2018, doi: http://dx.doi.org/10.1109/COMST.2018.2815638.

[315] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, May 2017, doi: http://dx.doi.org/10.1109/MCOM.2017.1600951.

[316] F. Malandrino, C. F. Chiasserini, G. Einziger, and G. Scalosub, "Reducing Service Deployment Cost Through VNF Sharing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2363–2376, Dec 2019, doi: http://dx.doi.org/10.1109/TNET.2019.2945127.

[317] M. Crippa, P. Arnold, V. Friderikos, B. Gajic, C. Guerrero, O. Holland, I. Labrador, V. Sciancalepore, D. Von Hugo, S. Wong, F. Yousaf, and B. Sayadi, "Resource sharing for a 5G multi-tenant and multi-service architecture," 2017, [Online] Available: https://bit.ly/2vhtKyq.

[318] J. Gang and V. Friderikos, "Optimal resource sharing in multi-tenant 5G networks," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2018-April, pp. 1–6, 2018, doi: http://dx.doi.org/10.1109/WCNC.2018.8377326.

[319] C. Vlachos, V. Friderikos, and M. Dohler, "Optimal Virtualized Inter-Tenant Resource Sharing for Device-to-Device Communications in 5G Networks," *Mobile Networks and Applications*, vol. 22, no. 6, pp. 1010–1019, 2017, doi: http://dx.doi.org/10.1007/s11036-017-0822-0.

[320] D. Panaitopol, A. Cipriano, K. Katsalis, N. Nikaien, C.-Y. Chang, F. Kuo, K. Kowalik, H. Kokkinen, G. Agapiou, and A. Kliks, "Spectrum and RAN sharing in 5G networks - a COHERENT approach," Oulu, FINLAND, 06 2017, [Online] Available: http://www.eurecom.fr/publication/5203.

[321] J. Gang and V. Friderikos, "Inter-Tenant Resource Sharing and Power Allocation in 5G Virtual Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7931–7943, Aug 2019, doi: http://dx.doi.org/10.1109/TVT.2019.2917426.

[322] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant mobile networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 662–675, April 2019, doi: http://dx.doi.org/10.1109/TNET.2019.2895378.

[323] H. Khalili, A. Papageorgiou, S. Siddiqui, C. Colman-Meixner, G. Carrozzo, R. Nejabati, and D. Simeonidou, "Network Slicing-aware NFV Orchestration for 5G Service

Platforms," pp. 25–30, June 2019, doi: http://dx.doi.org/10.1109/EuCNC.2019.8802048.

[324] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, "On multi-domain network slicing orchestration architecture and federated resource control," *IEEE Network*, vol. 33, no. 5, pp. 242–252, Sep. 2019, doi: http://dx.doi.org/10.1109/MNET.2018.1800267.

[325] R. Kunst, *A QoS-aware resources sharing architecture for homogeneous and heterogeneous wireless networks*, 2017, [Online] Available: https://bit.ly/2UPKxb9.

[326] P. H. Isolani, N. Cardona, C. Donato, G. A. Pérez, J. M. Marquez-Barja, L. Z. Granville, and S. Latré, "Airtime-based Resource Allocation Modeling for Network Slicing in IEEE 802.11 RANs," *IEEE Communications Letters*, pp. 1–1, 2020, doi: http://dx.doi.org/10.1109/LCOMM.2020.2977906.

[327] P. H. Isolani, J. Haxhibeqiri, I. Moerman, J. Hoebeke, J. M. Marquez-Barja, L. Z. Granville, and S. Latré, "An SDN-based framework for Slice Orchestration using In-Band Network Telemetry in IEEE 802.11," *IEEE Conference on Network Softwarization (NETSOFT2020)*, pp. 1–3, 2020.

[328] P. H. Isolani, N. Cardona, C. Donato, J. Marquez-Barja, L. Z. Granville, and S. Latré, "SDN-based Slice Orchestration and MAC Management for QoS delivery in IEEE 802.11 Networks," pp. 260–265, 2019, doi: http://dx.doi.org/10.1109/SDS.2019.8768642.

[329] J. Costa-Requena, R. Kantola, A. Y. Ding, J. Manner, Y. Liu, and S. Tarkoma, "Software Defined 5G Mobile Backhaul," 2014, doi: http://dx.doi.org/10.4108/icst.5gu.2014.258054.

[330] J. Costa-Requena, J. L. Santos, and V. F. Guasch, "Mobile backhaul transport streamlined through SDN," 2015, doi: http://dx.doi.org/10.1109/ICTON.2015.7193588.

[331] W. Kiess, M. R. Sama, J. Varga, J. Prade, H. J. Morper, and K. Hoffmann, "5G via evolved packet core slices: Costs and technology of early deployments," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, vol. 2017-October, pp. 1–7, 2018, doi: http://dx.doi.org/10.1109/PIMRC.2017.8292691.

[332] Y. Li, "Virtualization in Wireless Networks," *Wireless and Mobile Networking*, pp. 1–7, 2009, [Online] Available: https://www.cse.wustl.edu/{~}jain/cse574-14/ftp/virwn.pdf.

[333] M. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Comm Mag*, 2009, doi: http://dx.doi.org/10.1109/mcom.2009.5183468.

[334] A. Framework, "GS NFV 002 - V1.1.1 - Network Functions Virtualisation (NFV); Architectural Framework," vol. 1, pp. 1–21, 2013, available [Online]: https://bit.ly/2LwTxNa.

[335] S. Farhat, S. Lahoud, A. E. Samhat, and B. A. Cousin, "Resource Sharing in 5G Multi-Operator Wireless Network," *International Journal of Digital Information and Wireless Communications*, vol. 8, no. 3, pp. 156–161, 2018, doi: http://dx.doi.org/10.17781/p002431.

[336] G. Fortetsanakis, M. Papadopouli, G. Karlsson, M. Dramitinos, and E. A. Yavuz, "To subscribe, or not to subscribe: Modeling and analysis of service paradigms in cellular markets," *2012 IEEE International Symposium on Dynamic Spectrum Access Networks, DYSPAN 2012*, pp. 189–200, 2012, doi: http://dx.doi.org/10.1109/DYSPAN.2012.6478130.

[337] B. Leng, P. Mansourifard, and B. Krishnamachari, "Microeconomic analysis of base-station sharing in green cellular networks," *Proceedings - IEEE INFOCOM*, pp. 1132–1140, 2014, doi: http://dx.doi.org/10.1109/INFOCOM.2014.6848044.

[338] A. Georgakopoulos, A. Margaris, K. Tsagkaris, and P. Demestichas, "Resource Sharing in 5G Contexts: Achieving Sustainability with Energy and Resource Efficiency," *IEEE Vehicular Technology Magazine*, 2016, doi: http://dx.doi.org/10.1109/MVT.2015.2508319.

[339] L. Cano, A. Capone, G. Carello, and M. Cesana, "Evaluating the performance of infrastructure sharing in mobile radio networks," *IEEE International Conference on Communications*, vol. 2015-Septe, pp. 3222–3227, 2015, doi: http://dx.doi.org/10.1109/ICC.2015.7248820.

[340] J. Lun and D. Grace, "Software defined network for multi-tenancy resource sharing in backhaul networks," pp. 1–5, March 2015, doi: http://dx.doi.org/10.1109/WCNCW.2015.7122519.

[341] J. Zhang, J. Zhang, Y. Zhao, and X. Wang, "Time-dependent spectrum resource sharing in flexible bandwidth optical networks," *IET Networks*, vol. 1, no. 4, pp. 189–198, December 2012, doi: http://dx.doi.org/10.1049/iet-net.2012.0119.

[342] F. Marzouk, H. Touati, R. Alheiro, and A. Radwan, "Analysis of Multi-Operator Resource Sharing," pp. 269–273, Sep. 2019, doi: http://dx.doi.org/10.1109/5GWF.2019.8911698.

[343] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, no. 648382, pp. 55 765–55 779, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2872781.

[344] C. Bluemm, Y. Zhang, P. Alvarez, M. Ruffini, and L. A. DaSilva, "Dynamic energy savings in Cloud-RAN: An experimental assessment and implementation," pp. 791–796, May 2017, doi: http://dx.doi.org/10.1109/ICCW.2017.7962755.

[345] R. Shrivastava, S. Costanzo, K. Samdanis, D. Xenakis, D. Grace, and L. Merakos, "An SDN-based framework for elastic resource sharing in integrated FDD/TDD LTE-A HetNets," pp. 126–131, Oct 2014, doi: http://dx.doi.org/10.1109/CloudNet.2014.6968980.

[346] M. Ruffini, D. B. Payne, and L. Doyle, "Protection strategies for long-reach PON," pp. 1–3, Sep. 2010, doi: http://dx.doi.org/10.1109/ECOC.2010.5621156.

[347] M. Ruffini, M. Achouche, A. Arbelaez, R. Bonk, A. Di Giglio, N. J. Doran, M. Furdek, R. Jensen, J. Montalvo, N. Parsons, T. Pfeiffer, L. Quesada, C. Raack, H. Rohde, M. Schiano, G. Talli, P. Townsend, R. Wessaly, L. Wosinska, X. Yin, and D. B. Payne, "Access and metro network convergence for flexible end-to-end network design [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 6, pp. 524–535, June 2017, doi: http://dx.doi.org/10.1364/JOCN.9.000524.

[348] S. McGettrick, D. B. Payne, and M. Ruffini, "Improving hardware protection switching in 10Gb/s symmetric Long Reach PONs," pp. 1–3, March 2013, doi: http://dx.doi.org/10.1364/OFC.2013.OW3G.2.

[349] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Cellslice: Cellular wireless resource slicing for active ran sharing," pp. 1–10, Jan 2013, doi: http://dx.doi.org/10.1109/COMSNETS.2013.6465548.

[350] O. Narmanlioglu and E. Zeydan, "New era in shared C-RAN and core network: A case study for efficient RRH usage," pp. 1–7, May 2017, doi: http://dx.doi.org/10.1109/ICC.2017.7997428.

[351] B. Niu, Y. Zhou, H. Shah-Mansouri, and V. W. S. Wong, "A dynamic resource sharing mechanism for cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8325–8338, Dec 2016, doi: http://dx.doi.org/10.1109/TWC.2016.2613896.

[352] V. K. Choyi, A. Abdel-Hamid, Y. Shah, S. Ferdi, and A. Brusilovsky, "Network slice selection, assignment and routing within 5G Networks," pp. 1–7, Oct 2016, doi: http://dx.doi.org/10.1109/CSCN.2016.7784887.

[353] FCC. (2019, march) Federal Communications Commission: Millimeter Wave 70/80/90 ghz Service. [Online] Available: https://bit.ly/30w3u1D.

[354] ——, "Federal Communications Commission: Report and order and further notice of proposed rule-making," 2016, [Online] Available: https://bit.ly/2YypuY0.

[355] C. Both, R. Guimaraes, F. Slyne, J. Wickboldt, M. Martinello, C. Dominicini, R. Martins, Y. Zhang, D. Cardoso, R. Villaca, I. Ceravolo, R. Nejabati, J. Marquez-Barja, M. Ruffini, and L. DaSilva, "FUTEBOL Control Framework: Enabling Experimentation in Convergent Optical, Wireless, and Cloud Infrastructures," *IEEE Communications Magazine*, 2019, doi: http://dx.doi.org/10.1109/MCOM.001.1900270.

Chapter A

Resource sharing in end-to-end 5G networks

The work presented in the Appendix is our early work on studying concepts of resource sharing in 5G ecosystems, and as such it is part of the **Contribution 1** and it is based on:

N. Slamnik-Kriještorac, H. Kremo, M. Ruffini and J. M. Marquez-Barja, "Sharing Distributed and Heterogeneous Resources toward End-to-End 5G Networks: A Comprehensive Survey and a Taxonomy," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1592-1628, 2020, doi:10.1109/COMST.2020.3003818.

The widespread concept of *sharing* can be defined as a joint use of resources enabled by on-demand exchange, or by loaning of valuable goods [200]. Sharing brings its beneficial nature in many domains including social and economic systems, as well as in nature. Based on that fact, we anticipate and envision that provisioning of the models and methods for sharing of network resources represents one of the fundamental steps in designing FCNs. As a pioneer among FCNs, 5G represents the fifth generation of wireless technologies for digital cellular networks. Built upon 4G systems, 5G is an evolution considered to be the convergence of Internet services with legacy mobile networking standards leading to the mobile Internet over heterogeneous networks with high-speed broadband [201].

The main focus of our survey is on the applicability of sharing in the context of FCNs with the goals to: i) present current trends in sharing of network resources, ii) provide research community with knowledge on the existing sharing techniques, iii) outline the challenges in the implementation of these techniques, and finally and most importantly iv) provide a taxonomy which brings the main features of a comprehensive sharing model into focus, facilitating the creation of models suitable to build more efficient FCNs.

Given Fig. A.1, the aforementioned comprehensive sharing models span both physically tangible and intangible types of network resources (pool of shareable resources) in the *wireless* as well as in the *optical domain*, altogether with *IoT*, *network edge*, and *cloud domains*. Moreover, sharing actors (Fig. A.1, e.g., network operators (i.e., MNOs), Service Providers

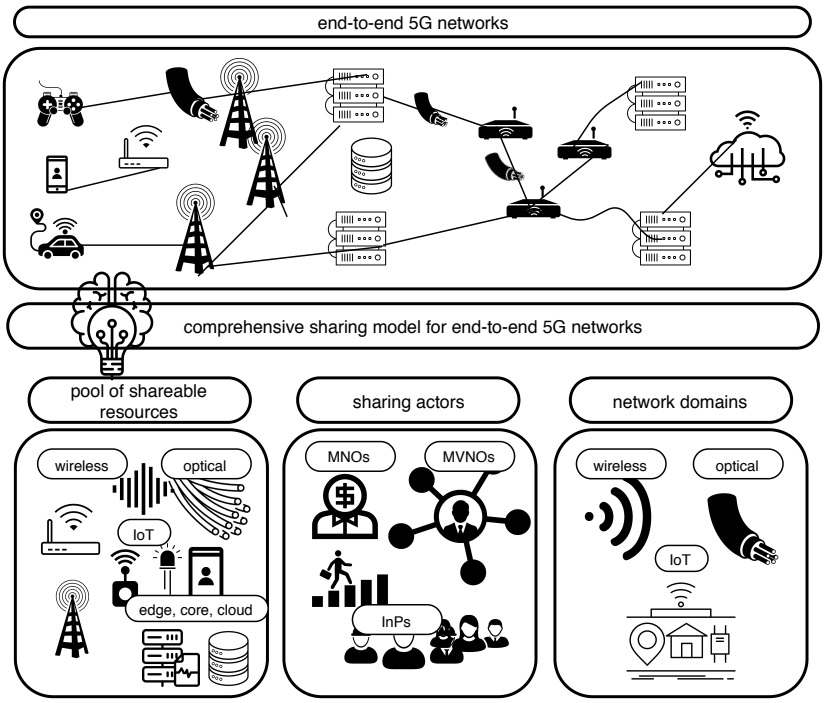


Figure A.1: Extensive and comprehensive sharing of distributed and heterogeneous resources.

(SPs), InPs, Mobile Virtual Network Operators (MVNOs), and users) can increase both revenues and users' satisfaction with their services if they share resources from an end-to-end perspective. Such concept is shown in Fig. A.2, illustrating it from a 5G network perspective.

Due to the strong heterogeneity in terms of network resources and technologies, it is utmost important for such sharing models to have an *end-to-end perspective* of 5G networks. As shown in Fig. A.2, the scope of a 5G network as an FCN spans different network segments, such as users' domain (i.e., UE), RAN, edge (depending on the deployment, it can be part of RAN, with edge servers deployed within Base Transceiver Stations (BTSs)), and finally core, and cloud. Furthermore, each of these network segments are deployed/developed/hosted by different InPs, operators, manufacturers, SPs, etc. (represented by colored boxes in Fig. A.2), making a 5G network resourceful but highly heterogeneous ecosystem. Besides different network segments starting from user domain all the way to the cloud, 5G takes advantage of different communication technologies such as wireless and optical, as shown in specific network segments. Finally, it includes a wide variety of IoT devices that are key components in Industry 4.0 or Smart cities domains. Hence, a vast end-to-end perspective of 5G network is a cohesion of wireless and optical technologies, connecting IoT and non-IoT devices from user domain to the core and cloud, taking advantage of edge computing which aims at reducing the overall end-to-end latency by exposing resources to the network edge. Therefore, the end-to-end perspective in the context of resource sharing means that the design and the optimization of networks should be achieved by sharing a wide variety of resources, starting with users' domain, through RAN and edge towards core network and

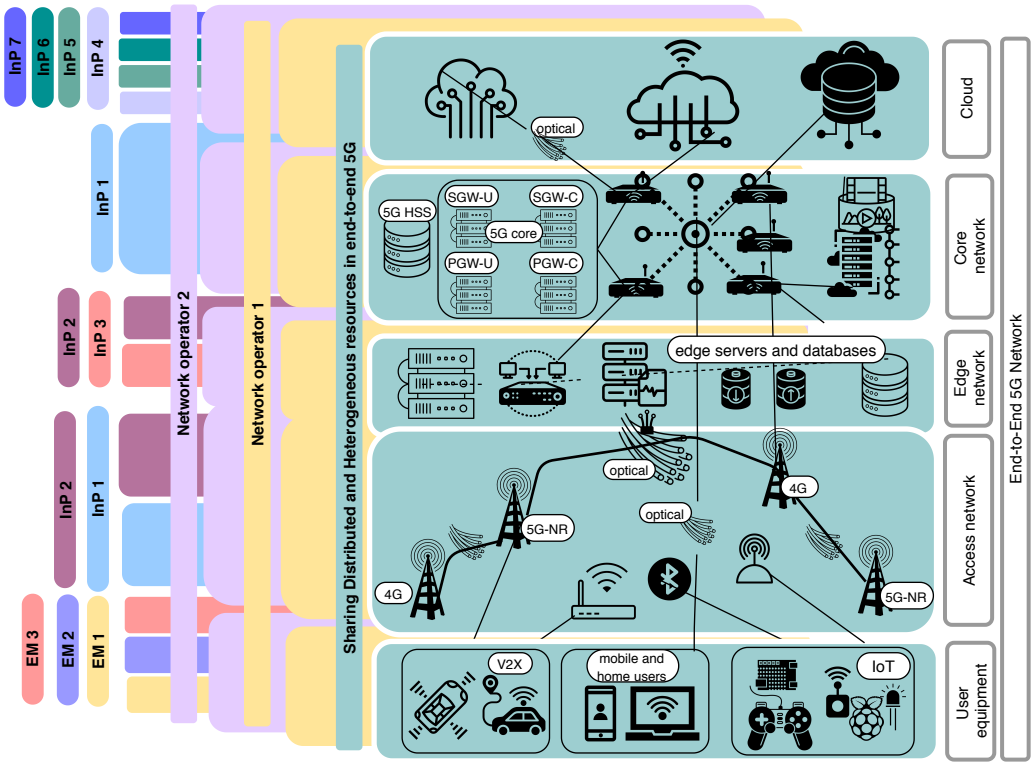


Figure A.2: The End-to-End 5G networks perspective starting from user domain, through access and edge networks, towards core and cloud (EM, InP).

cloud. In this way, instead of having a full ownership of the specific network segment, all of the sharing actors make their pool of resources available for sharing.

In Fig. A.2, we also use a comprehensible color code followed by explanatory boxes (e.g., Network operator 1, Infrastructure provider 2, Sharing Distributed and Heterogeneous resources in end-to-end 5G, etc.), clearly differentiating scenarios:

- in which all resources from user domain to the cloud, including wireless, optical, and IoT, are shared (green color in Fig. A.2),
- and those where different network segments (i.e., edge network, access network, etc.) are supplied, maintained, and/or owned by different parties (e.g., network operators, infrastructure providers, and equipment manufacturers) (other colors in Fig. A.2).

Sharing provides tremendous benefits regardless of the environment in which it is applied, and its benefits are especially known in economics. In the sharing economy, the participants (i.e., sharing actors), share and use valuable items like cars or houses without the need for exclusive ownership [200]. At the same time, sharing creates opportunities for others to extract value from idle possessions or talents [202].

In the emerging sharing cities paradigm [203] - including increasingly popular smart cities -

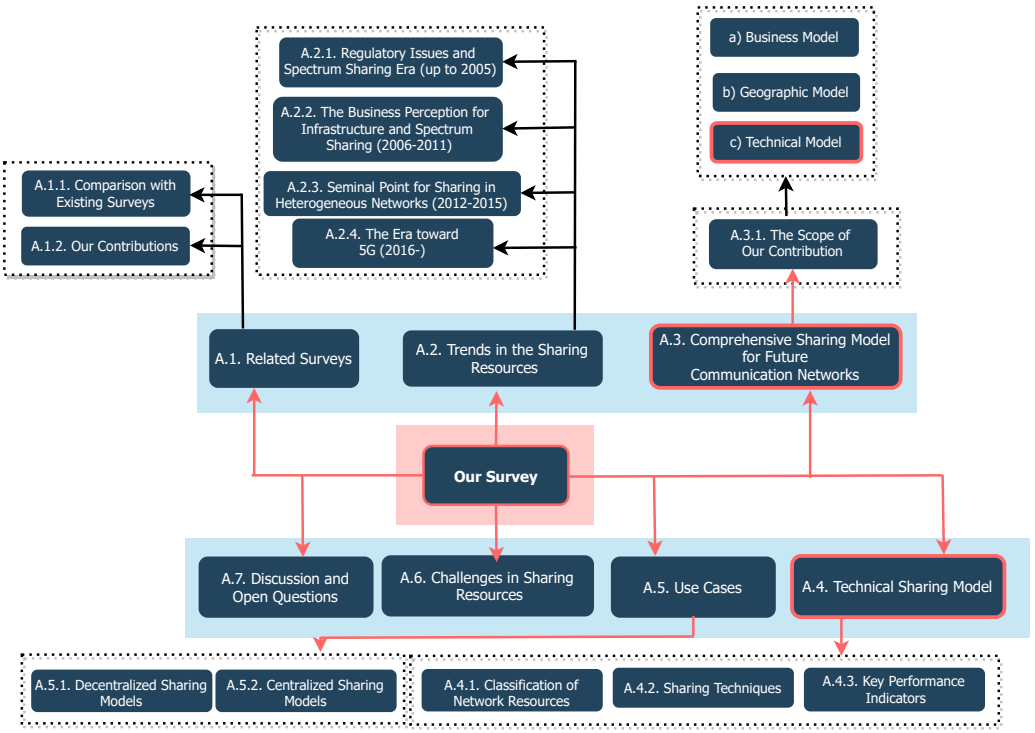


Figure A.3: Organization of Appendix A.

goods such as spaces or venues for collaboration, parking spots, and publicly owned handy bikes are shared. For example, the National Industrial Symbiosis Program (NISP) is a model to optimize use of resources in commercial business and move toward circular economy through sharing. In an eight year period, in Europe and around the world, NISP has helped businesses to: i) save £1 billion in costs, ii) generate £993 million in additional sales, iii) safeguard over 10,000 jobs, iv recover and reuse 38 million tons of material, v) reduce 39 million tons of industrial carbon emissions, and vi) save 71 million tons of industrial water [200].

Another interesting example comes from the microscopic world, in which the same species of bacteria compete for the same resources when living in homogeneous communities. Such competition results in their decreased growth. However, when they change their feeding habits to share the resources more effectively by coexisting in mixed communities with other species and by reusing each other's waste products, the operation and well-being of the whole heterogeneous community is greatly improved [204]. With the introduction of 5G now is the right time to look up to such fascinating examples [205], and to exploit the resource sharing potential of communications networks.

Complementary to sharing of goods, network sharing is a paradigm which embraces a set of strategies that enable network operators to use their resources jointly in order to reach their common goal: to provide and guarantee user services while achieving energy and cost reduction [206]. As an illustration of the benefits of such sharing, Bousia et al. [206] report considerable improvement (increased energy efficiency by 174% and cost reduction by 86%),

when the number of operators who share their underutilized network elements increases from four to six.

Moving our focus to the digital world, there is a prediction in Cisco Forecast and Trends paper [3] that an ever-increasing number of devices that are wirelessly connected to the Internet (smart-phones, tablets, IoT devices, etc.), will reach approximately 12.3 billion by 2023. As a consequence, such growth unavoidably leads to tremendous increase in service requests for applications like video, interactive gaming, M2M communications, etc. In the 5G community these applications fall into the three main areas: mMTC, uRLLC, and eMBB [4]. Applications falling into these categories impose highly specific and stringent QoS requirements. For the network operators, these QoS requirements are then tied to provisioning of different network resources. Consequently, the excessive growth in service requests becomes a heavy technological and economic burden for the operators.

From the purely technical perspective, once the service request arrives, the pool of heterogeneous and distributed resources is invoked. Then, selection and chaining of adequate portions of the network resources is performed. These resources are then provided to the service which initiated the request. The resources are carefully selected from the resource pool and customized to the service request. However, these resources are not localized within a centralized pool. In reality the resources are widely (geographically) distributed across the entire network (Fig. A.1, and Fig. A.2). The conflict between widely disseminated network infrastructure and its strict ownership boundaries clearly and urgently presses to create and implement new sharing models for the network resources. Several important points should be emphasized:

1. In such dynamic and challenging environment as 5G, it is essential to enable coexistence of diverse existing services and facilitate easy creation of new ones [207].

2. When all network operators have static amount of dedicated resources, a significant percentage of those resources can go to waste if the excess is not shared among the operators [207]. Hence, once the heterogeneous network resources are not needed, they should be released for sharing and temporarily given to other entities.

3. Operators should rethink their traditional business models, evolving from owning all the resources (from very intangible items like spectrum to physically tangible ones like electronic equipment, radio masts, and towers) to sharing of these resources [208]. However, a corresponding model made of rules for sharing (such as the operators' business model) should be established and used wherever and whenever sharing is an option.

   While formal business models are out of scope of our work, we want to provide the research community with an extensive overview and knowledge base of resource sharing that will enable future dynamic network environments. The importance of the aforementioned approach is also emphasized in Fig. A.3, where the red-framed Section A.3.1 elaborates the comprehensive sharing model and its features, with a specific focus on the technical sharing model in Section A.4.

4. The advent of emerging technologies, such as SDN, and NFV provide momentum for new design principles toward software-defined 5G networks that are expected to facilitate resource sharing, and resource management in general. The aforementioned

is viable since virtualization is a technique that abstracts network resources, making them independent from the underlying physical infrastructure. On the other hand, SDN simplifies resource management by decoupling the control and data, positioning them into two distinct planes via logically centralizing network intelligence. As both SDN and virtualization are recognized as crucial enablers for network sharing, we elaborate on their impact on resource sharing in Section A.4.2.

Observing how resource sharing has evolved over time, one can recognize the transition from only hardware-based sharing to overall softwarization, which is discussed in greater detail in Section A.2. This specific transition from hardware-based to software-based sharing evolved into different models that at first identify and distinguish all shareable resources, and then offer them for sharing. Our perspective on this shift toward softwarization will pave the way for new contributions in diverse research domains, such as dynamic network configurations and slicing, new service creation and delivery, and techno-economics.

The overall organization of this work is presented in Fig. A.3, which clearly shows the structure of the sections, briefly announcing the content related to each of them. Within Section A.1, we present related work by comparing our views to other related surveys. Then, based on that comparison, we specify the contributions which our survey provides to the research community. Importantly, Sections A.2 and A.3 address resource sharing from two different viewpoints: one showing the evolution of sharing over time, and another presenting dimensions of the sharing model that have to be carefully considered and designed prior to sharing. In particular, the trends in resource sharing over the period of the last 20 years are discussed in Section A.2. Section A.3 defines the position of the sharing paradigms in a generalized end-to-end FCN architecture, providing an in-depth taxonomy of this area. The taxonomy brings relevant features of sharing models into the focus, pointing at all the dimensions that have to be carefully designed and synchronized in order to create more efficient FCNs. It presents a hierarchical view of the issues and solutions, per model: business, geographic, and technical; and per layer: infrastructure, orchestration, and service. In Section A.5, we present specific use cases which exemplify the resource sharing. After this, section A.6 reports the main research challenges that need to be taken into consideration during careful design of any sharing model. A baseline for open research questions and the following discussion is presented in Section A.7.

A.1 Related Surveys

In this section, we present an analysis of the existing surveys available in the literature addressing sharing-related topics. Moreover, we highlight the new and complimentary contributions that our survey brings to the research community. Fig. A.4 provides our insight into the classification of existing surveys on sharing for next generation communication networks. The figure illustrates lack of an overall end-to-end approach in the research community. The analyzed work only considers specific parts of the network infrastructure, such as spectrum sharing in wireless networks.

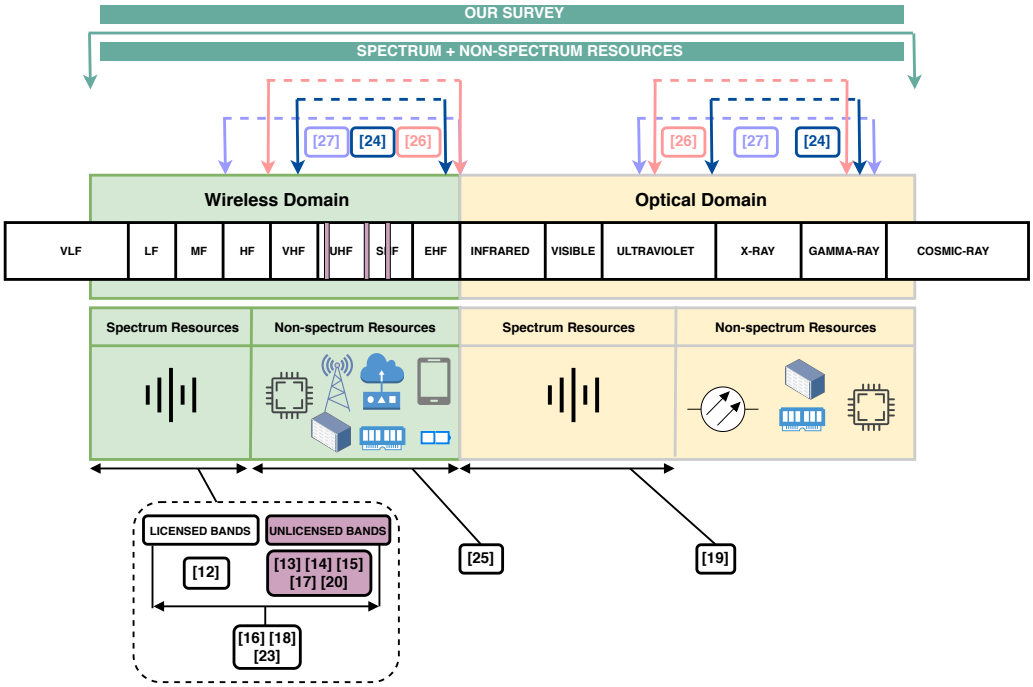


Figure A.4: Overview of existing related surveys.

A.1.1 Comparison with Existing Surveys

This section provides insight into current research through the analysis of existing survey papers on the topic of sharing resources in end-to-end next generation communication networks (Fig. A.4). Our approach takes into account a challenging end-to-end overview of FCNs, considering surveys in both wireless and optical domains, and including IoT, edge, and cloud.

Sharing of Radio and Optical Spectrum According to prior surveys [209, 210, 211, 212, 213, 214, 215, 216, 217], sharing of resources in communication networks usually entails spectrum as the bottleneck commodity with the highest demand and the smallest availability. The imminent shortage of this type of resource, coupled with the increasing demand for higher capacity, is a strong motivation for researchers to study practical solutions for efficient spectrum sharing. During the last 15 years, and more recently with anticipated deployment of 5G wireless networks [209], the interest for spectrum sharing has grown even larger, resulting in a vast number of publications investigating and presenting new sharing solutions for this intangible resource. As Fig. A.4 shows, under the roof of the wireless networks and depending on the spectrum ownership, the existing surveys address spectrum sharing in: i) *licensed* bands, ii) *unlicensed* bands, and iii) *both*. Tehrani et al. [209] study the main concepts of dynamic spectrum sharing and different sharing scenarios, with the focus on practical solutions which efficiently utilize scarce licensed bands in a shared manner. They also recognize and present the major challenges related to sharing in licensed parts of the wireless spectrum. With respect to unlicensed bands, the Cognitive Radio (CR) has received

the prominent attention [210, 211, 212, 214, 217].

The CR paradigm addresses the issue of spectrum scarcity and underutilization by enabling a technique called Dynamic Spectrum Allocation (DSA), which allows users to opportunistically access unlicensed bands [210]. The most general classification of CR network paradigms is given by Goldsmith et al. [217] as follows: i) *underlay*, ii) *overlay*, and iii) *interweave*, characterized by the rule cognitive users follow in their operation. Furthermore, Nair et al. [210] provide a comprehensive overview of the use of game theory as the enabler for DSA. A survey on full spectrum sharing in CR networks, but with main focus on its implementation in 5G networks, is presented by Hu et al. in [211]. The authors discuss further expansion of the spectrum range (from 1 GHz to 100 GHz), motivated by the demand to meet all the critical service requirements in 5G networks, such as wider coverage, massive capacity, massive connectivity, and low latency. Similarly to the approach adopted by Nair el al. in [210], the problems with spectrum allocation are discussed under the game theory umbrella in [211]. Beside other spectrum sharing schemes, such as: Device to Device (D2D) spectrum sharing, In-Band Full Duplex (IBFD), NOMA, LTE-Unlicensed (LTE-U)-based spectrum sharing, Zhang et al. also present CR as an intelligence layer on top of the aforementioned approaches, first in a specific IoT context in [212] and then as an advanced technique for spectrum sharing in 5G networks in [214].

When considering spectrum sharing across both wireless and optical bands, it can be observed that these are typically utilized by a diverse pool of wireless devices [213, 214], rather than single-technology devices. Their difference in terms of technologies and traffic requirements implies interaction across technologies, which is gaining momentum [213]. Voicu et al. address spectrum sharing mechanisms for wireless inter-technology coexistence (e.g., WiFi/Long-Term Evolution (LTE), WiFi/blacktooth, LTE/D2D or Narrowband Internet of Things (NB-IoT)), surveying both technical and non-technical aspects. As non-technical aspects that are the most influential on the design of the spectrum sharing mechanisms, they identify the business models and the social practices. The authors observe that sometimes the best technical solutions for sharing may not be adopted due to non-technical concerns like the lack of agreement among sharing participants [213]. For instance, the primary spectrum owners must be incentivized to yield exclusive spectrum rights [218, 219].

Regarding the heterogeneity of 5G networks, Zhang et al. [214] present the idea to study multiple spectrum sharing techniques jointly, in order to provide a global spectrum sharing approach which encompasses multiple radio technologies. In order to better discern the concept and all the practicalities of spectrum sharing in upcoming 5G networks, a profound understanding of spectrum sharing in LTE is a must. To that goal, Ye et al. [215] present the overview of LTE spectrum sharing techniques, with the focus on three spectrum segments: i) TV white space channels, ii) frequently unused service-dedicated 3.5 GHz, and iii) 5 GHz unlicensed band [215]. Finally, Ahmad et al. present a thorough review of recent advances in spectrum sharing in 5G networks [220]. *However, all of the above-mentioned surveys solely tackle the wireless domain*.

In optical communications networks, spectrum is by itself not a scarce resource, as each individual fiber strand can carry several Tbps of capacity. In addition, optical transmissions networks are typically closed systems in two ways. Firstly there is usually only one operator running services over each fiber pair; secondly, optical systems are mostly deployed using technology from a single vendor. However, recently the trend is changing, as the possibility

to open up an optical system to operate with components from more than one vendor is being investigated across several industry-drive consortia (most notably, the Optical Networking Foundation (ONF), the Open ROADM Multi-Source Agreement, which defines interoperability specifications for Reconfigurable Optical Add/Drop Multiplexers (ROADM), and the Telecom Infra Project (TIP)). Considering also that using additional optical fibre is expensive, especially in long-haul links that require the addition of several in-line amplifiers, the concept of fiber spectrum sharing has been recently explored, especially with the rise of Elastic Optical Networks (EONs). Spectrum management techniques for EONs were recently addressed in [216] by Talebi et al., where they show, for example, the importance of efficient spectrum sharing across backup optical paths.

Sharing of Resources Other than Spectrum Virtualization is recognized as a technique that enables efficient resource sharing among different operators, services, and applications [18, 221, 222]. According to Kliks et al. [18], the broad idea of virtualization is that it enables separation of services or service requests from the actual resources. Considering non-spectrum resources (although confined only to the wireless domain) Zahoor and Mir [221] present the survey on virtualization in the context of IoT resource management, providing the insight into how IoT infrastructure can be virtualized in order to be shared. Here we elaborate on several publications, which study both wireless and optical domains. For instance, Bianzino et al. [222] depict the key paradigms, including virtualization of the FCN infrastructure, which can be exploited to reach network "greening" (i.e., reduction of energy consumption). Although Bianzino et al. [222] mainly consider the wired domain, they also outline insights on how to deploy the paradigms they introduce in the wireless domain. Mamushiane et al. in [223] offer an overview of the concept of SDNs as an enabler of sharing, together with an assessment of its impact on CapEx and OpEx. In particular, CapEx includes all expenses related to the initial investments that the operators face during equipment purchase and installation. On the other hand, OpEx is related to the network maintenance and other expenses which are necessary for proper operation of the communication network on a daily basis. Mamushiane et al. [223] tackle both the optical and the wireless domain, and with respect to the optical domain they investigate how sharing of the active backhaul through softwarization reduces both costs.

Finally, Kliks et al. [18] provide a comprehensive study of all the perspectives for resource sharing in 5G networks, considering both the wired and the wireless domains. Regardless of the fact that the above reference is not a survey, but rather a literature overview, it is one of the rare attempts to examine resource sharing in 5G networks from a broader perspective. Hence, it presents an overview of the concepts for 5G implementation in a flexible and programmable manner through virtualization. Most notably, the authors provide a generalized architecture for FCN, but only in the context of sharing resources in the wireless domain. We adopt and expand their architecture and try to exploit it in a broader sense by surveying network sharing from an end-to-end perspective, and in both wireless and optical domains, while also including IoT, edge and cloud resources.

A.1.2 Our Contributions

From the previous section we conclude that the existing surveys do not cover sharing of network resources from the end-to-end standpoint. In particular, Fig. A.4 depicts a quite unbalanced scenario, where the majority of the surveys solely tackle spectrum sharing in wireless networks. To the best of our knowledge, this work is the first attempt to research and survey network sharing in an end-to-end manner, thereby considering heterogeneous network resource sharing that crosses both the wireless and optical network domains, and extends to IoT, edge, and cloud paradigms, which altogether coexist and define the 5G network. The impact of our survey is in providing an extensive taxonomy on the sharing of heterogeneous resources in FCN with a viewpoint that goes beyond the boundaries between networks and operator domains. We examine the sharing potential of all resources from users' domain, RAN, edge, and core network. Gathering information on how resources of these separate domains used to be shared, and up to what extent, as well as determining the similarities between sharing models, can help us understand the true potentials of each technology and domain.

Thus, the overall impact of this survey consists of the following contributions:

1. Helping the research community to identify up to what extent can network resources be shared, answering the questions on what could be and what should be shared.
2. Presenting the existing use cases and techniques used to enable sharing of distributed and heterogeneous resources.
3. Recognizing and presenting sharing challenges and requirements arising from the highly dynamic, heterogeneous, and highly diverse 5G environment.
4. Providing a taxonomy which will help researchers design new sharing models, by thoroughly investigating current network sharing challenges.

Another contribution of this survey is that it will provide a solid reference for researchers willing to address the following topics:

- Creating a flexible environment as enabler for new diverse services for the end users.
- Implementing comprehensive sharing model in real-life scenarios, which includes collaboration with group members working on NFV and softwarization.
- Developing techno-economic models for sharing.
- Extending and enhancing existing sharing approaches by leveraging the AI umbrella.

A.2 Trends in the Sharing Resources

This section describes how the concept of resource sharing has evolved over time and classifies publications both by time and topic, which we summarize in Fig. A.5. Studying these

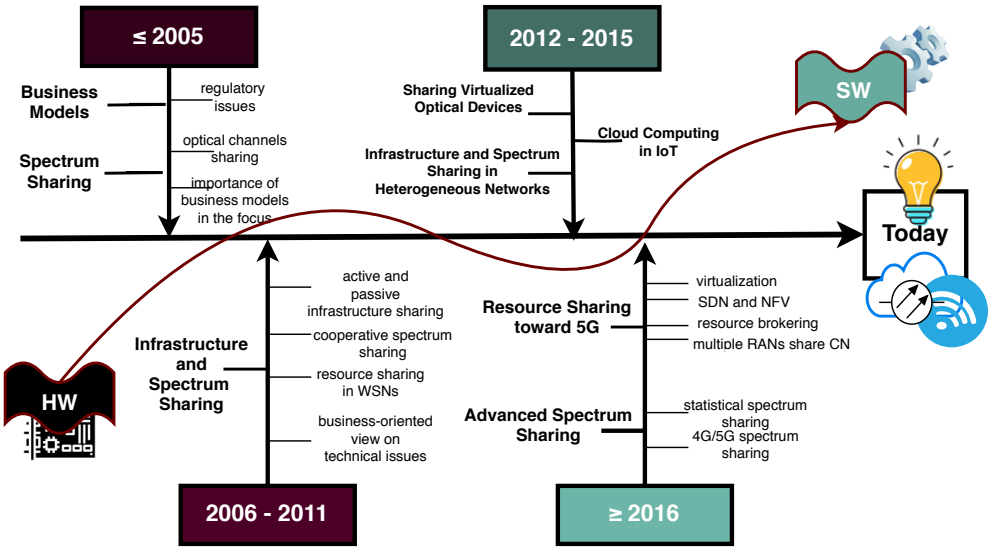


Figure A.5: Timeline of Resource Sharing; Sharing Trends.

topics, we identified that sharing resources in wireless and optical domain [224, 225] have always been considered separately, despite the fact that at times they used similar techniques [7, 11, 226]. One of the most important tendencies, not only in resource sharing but also in computing, can be recognized from the illustrated timeline. Namely, the tendency to share physical resources has changed over time, following the emerging popularity of ubiquitous techniques such as virtualization and software defined networking. Thus, it can be clearly observed that in the early 2000 s and even before the trend was to share physical resources (i.e., hardware), while recent trend is to share logical resources which are the result of softwarization/abstraction of physical resources. If the tendencies related to the specific types of resources are taken into consideration, one can primarily notice that spectrum has always been considered a bottleneck. Supported by the fact that spectrum is an enabler of wireless communication, it is not surprising that it still receives considerable attention among researchers [227, 228, 229, 230]. The following sections go into detail of the different phases, shown in Fig. A.5, taking into account a vast pool of heterogeneous and distributed network resources. In this section we briefly introduce these trends, which are then further elaborated in subsequent sections, where we describe use cases, sharing techniques, and more specific challenges.

A.2.1 Regulatory Issues and Spectrum Sharing Era (up to 2005)

One of the first attempts to approach spectrum sharing is presented by Gould and Kelleher [231], addressing the issue of frequency sharing between broadcasting satellites and other radio communications systems. This approach is followed by Prosch's in [232], which showed the possibility to increase spectrum efficiency by 30% when the Very High Frequency (VHF) spectrum band (30-300 MHz) is shared between the FM radio band (88-108 MHz) and the digital audio broadcast. Furthermore, an interesting analysis of interference caused

by multiple uncoordinated low-power transmitters for wireless network access towards fixed Point-to-Point (P2P) microwave receivers is given by Varma et al. in [233]. They determined and discussed the factors which directly impact the density of such uncoordinated users.

However, spectrum sharing was not emerging solely in the wireless domain, but also in optical, where for example Tridandapani and Mukherjee [234] examined channel sharing techniques in multi-hop optical networks. Another example of spectrum sharing is provided by Foschini in [235], which investigated the possibility of sharing optical bands among large numbers of high-speed users. In early 2000, Papadimitratos et al. [236] proposed an overlaid ad-hoc secondary network to share underutilized bandwidth resources in the primary cellular system. Here the authors also defined the Medium Access Control (MAC) protocol which enabled such scenario.

In this early phase, before 2006, a step further from spectrum sharing is provided by Ali in [224], who recognized optical node device as the dominant cost factor in overall backhaul network. At the same time, other researchers were exploring regulatory issues and the necessity for suitable business models. Beckman and Smith [237] identified regulatory issues as the crucial part for their feasibility study of resource sharing. Moreover, the importance of adequate business models for shared wireless networks is emphasized by Hultell et al. [238]. They recognized the need for a technical sharing framework, which enables sharing between multiple operators and service providers with strong focus on SLAs. The end of this era ceases with a critical review of controversial regulation rules provided by the U.S. Federal Communication Commission (FCC) [239], regarding the regulatory framework for sharing of landline access. In his review, Jones [240] provides a criticism towards regulations that fixed the price for access to the incumbents' switching facilities only for local voice service, while the price for accessing broadband equipment was left negotiable.

A.2.2 The Business Perception for Infrastructure and Spectrum Sharing (2006-2011)

At the beginning of the next period in our resource sharing timeline, CR started gaining momentum as a new Software Defined Radio (SDR) approach to radio spectrum sharing. The fixed spectrum assignment policies unavoidably led to unacceptably low spectrum utilization [241, 242]. With this in mind, Akyildiz et al. [241] presented one of the first concise overviews of all the characteristics of the CR concept and enabling technologies. Later, Akylidiz et al. surveyed the topic of spectrum management in CR networks, identifying developments and open research questions with focus on CR deployment without the need for modifying existing networks (i.e., primary spectrum owners) [243].

In that period, from 2006 to 2011, CR along with other enabling technologies like the software radio, spectrum sensing and mesh networks, was considered capable to facilitate new forms of spectrum sharing that could considerably improve spectral efficiency and alleviate scarcity [244]. However, any new technology would have no or little impact if inconsistent with spectrum policies, regardless of the opportunities and benefits it could bring. Accordingly, Peha in [244] discussed regulatory policies as the ultimate enablers for these emerging technologies, which can further facilitate spectrum sharing and increase spectrum utilization.

Furthermore, the importance of the CR paradigm was corroborated by many other papers, related to dynamic spectrum leasing [245], cooperative spectrum sharing [246], and opportunistic spectrum sharing in cognitive Multiple Input Multiple Output (MIMO) wireless networks [247]. In [248], while pointing at the opportunities and challenges in sharing the mostly underutilized government spectrum with private users, Marcus claimed again that research in this period was highly dependent on business and regulatory domains. Other business-oriented perspectives are provided by Frisanco et al. [2] and Meddour et al. in [249], but for infrastructure sharing. The authors studied both technical and business-related challenges in infrastructure sharing within the multi-vendor landscape of mobile communication networks.

One of the first attempts to apply sharing in Wireless Sensor Network (WSN) is presented by del Cid et al. in [250], aiming to resolve issues on concurrent use of WSN services, which leads to excessive contention of sensor node's resources for radio channel access. Also, Shi et al. in [227] studied resource management in IoT networks, from the perspective of scarce and non-renewable spectrum. Regarding the optical domain, an example of sharing is adopted and presented by Darcie et al. in [251]. They explored wavelength sharing on Passive Optical Networks (PONs) through the Wavelength Divison Multiplexing (WDM). This approach enables variable degrees of wavelength sharing by combining different wavelengths from multiple PONs.

A.2.3 Seminal Point for Sharing in Heterogeneous Networks (2012-2015)

The period from 2012 to 2015 has a significant impact on todays' research, since it includes studies of sharing of heterogeneous networks, providing a crucial asset for further research. Many technologies which are widely utilized for sharing toward FCNs were developed during this phase. We identify this period in our timeline (Fig. A.5), as a potential cornerstone for exploiting sharing of many different types of resources, including spectrum. Accordingly, Kibilda and DaSilva in [252] introduced the so-called *Networks without Borders*, as a mode of sharing infrastructure among both homogeneous and heterogeneous networks. Further advances in spectrum sharing are presented by Jorswieck et al. [253] and Park et al. [254]. Despite all the technology advances which reflect positive feedback from spectrum sharing [253], Park et al. point at severe security and privacy problems that have arisen as a consequence of sharing. Focusing on the framework of CR, they accentuated the importance of these problems, reviewing some of the critical security and privacy threats that impact spectrum sharing and its outcomes. These issues are classified into two categories: threats to sensing-driven spectrum sharing (such as PHY-layer threats, MAC-layer threats, and cross-layer threats) and threats to database-driven spectrum sharing (i.e., database interference attacks and threats to database access protocols) [254]. Furthermore, as for mission-critical types of services such as PPDR (e.g., FirstNet[1]) it is essential to ensure required spectrum resources to guarantee uninterrupted service. However, The Critical Communications Association (TCCA)[2] raised importance of spectrum sharing, since dedicating spectrum resources

[1]The First Responder Network Authority, or the FirstNet Authority, is an independent agency within the U.S. Department of Commerce's National Telecommunications and Information Administration (NTIA) that oversees FirstNet, the nation's communications network dedicated to emergency responders and the public safety community.

[2]The Critical Communications Associations: `https://tcca.info/about-tcca/`

to PPDR services would lead to underutilization, where an optimal solution would be to give highest priority to PPDR services but when they are not using the frequency, it should be leased to other types of services [255].

In spite of the ubiquitous popularity of virtualization techniques and SDN in todays' wireless networks, the first attempts to virtualize network resources occurred made in the fixed network domain. For instance, De Leenheer et al. [256] introduced sharing bandwidth among virtual optical networks grouped into clusters, followed by Vilalta et al. who introduced the concept of a virtual optical network resource broker [257]. Along with the popularity of virtualization techniques, the key enabling technique for the next generation of optical networks - Software Defined Optics (SDO) was introduced in [258, 259]. Wang et al. in [260], as well as Khandaker et al. in [258, 259], studied the concept of statistical spectrum sharing in the optical domain, enabling switching between base and peak rates through SDO. Many other researchers recognized the potentials in sharing optical devices [260, 261, 262] and in cooperative spectrum sharing [263].

As the final point in this section, we recognize the trends related to the IoT ecosystem, which belongs to the heterogeneous communication networks area. Heterogeneity of resources is not specific to IoT, but it appears to be most challenging in this domain due to the wide range of different devices, network connectivity options, communication protocols, communication methods, and so on. Hence, Silva et al. presented their attempt to bridge the heterogeneity among devices and to take advantage of it by symbiotic sharing between constrained IoT devices and unconstrained cellular devices [264]. At the same time, Kliem and Kao [265] applied the cloud computing paradigm to the management and sharing of resources in IoT, providing system design guidelines for specific use cases.

A.2.4 The Era toward 5G (2016-)

5G networks are supposed to offer new spectrum in the milimeter wave (mmWave) bands [266, 267], which can potentially move focus away from spectrum sharing. However, the deployment of services on such high frequencies has to be studied with attention, especially because of several open challenges. In accordance to that, Wan et al. [268], Al-Khatib et al. [15], and Shah et al. [24] briefly discuss spectrum sharing towards 5G, presenting the idea to reuse existing LTE spectrum together with new frequency bands used by 5G NR.

In the period from 2016 onwards, we find many publications focusing on virtualization of resources [16] empowered by SDN and network programmability [104] (Fig. A.5). This statement is supported by various references in both wireless and optical domains, which discuss sharing opportunities arising from virtualization and SDN. As one of the examples from the wireless domain, in [7] Zhang presents a wireless virtualization scheme, which offers abstraction and slicing as the base for their virtual network slicing/sharing framework. In particular, network slicing is a network concept that represents the whole network as a set of complete logical virtual networks, i.e., network slices, based on the physical shared infrastructure that is allocated to meet QoS demands [269, 270, 7, 271]. Extracting the potential from recent advances in SDR, SDN, and NFV, InPs can create virtual networks customized to the specific QoS requirements for different tenants, deploying application-driven network slicing [272]. In their work, Han et al. [272] propose a system for orchestrating resources and

services in heterogeneous networks that leverage SDN-supported network virtualization, and NFV-based MEC to realize application-driven end-to-end slicing. Providing a programmable SDN switch for flexible virtualization of radio resources by creating/removing virtual WiFi access points with dynamic bandwidth allocation, the work Han et al. presented in [272] can serve as a guideline for practices in radio resource sharing, enabled by network virtualization. Furthermore, Rawat [273] has introduced the concept of wireless virtualization as a technology that enables infrastructure sharing to multiple MVNOs, being considered as the best alternative to cognitive radio networks since it improves spectrum utilization efficiency, wireless network capacity, and coverage, with a special focus on wireless security (discussed in Sec. A.6). Thus, the sharing framework presented in [273] enables moving/switching users from one virtual network to another using hand-off techniques while maintaining a secure connection.

Fog computing and MEC are the promising network paradigms that bring cloud resources closer to the end-users, i.e., to the network edge [274, 273], in order to decrease the end-to-end latency. Altogether with NFV and SDN, edge computing is gaining significant attention recently, and represent an inevitable component of 5G networks. Therefore, an interesting sharing scheme where fog nodes share spare edge resources to help pre-process raw data of applications hosted in the cloud is presented in [274]. Under the decision control from an SDN controller, the volume of application data for pre-processing at the network edge is dynamically adjusted by using resources from all fog nodes.

In parallel, Afraz et al. [226] discuss how PON virtualization techniques introduced in [275, 276], together with SDN, impact the optical domain in terms of enabling multi-tenancy. Also, the role of a resource broker from Zhang's [7] and similar approaches used in the wireless domain, is replaced by a global orchestrator which orchestrates radio and transport resources jointly in Centralized Radio Access Network (C-RAN) using optical backhaul and fronthaul. The optical C-RAN is a centralized RAN with an optical transport whose wavelength resources can be dynamically shared among multiple BTSs [277]. Accordingly, significant contributions to the research community are provided by Marques et. al., Domincini et. al., Alvarez et. al., and Slyne et. al. since their work represents the integration of wireless and optical domain, enabled by SDN and virtualization of different wireless, optical, and edge/cloud resources [278, 279, 280, 281].

In addition to the huge increase in popularity of SDN in both wireless and optical domains, Municio et. al. [282] present the "Whisper" architecture, as an enabler for SDN-based IoT networks. The Whisper is a centralized SDN controller of a network which remotely controls nodes' forwarding and cell allocation. In line with the increase in IoT deployment and in the overall usage of IoT devices, sharing of IoT resources has become an immensely popular research topic in this period. For instance, Kouvelas et al. in [283] introduced an interesting theoretical foundation for resource sharing among IoT devices by exploiting graph theory. Furthermore, Yildirim and Tatar [284] present the two ways of sharing resources in WSNs: WSN virtualization and Middleware Based Server Systems (MBSSs), and discuss all advantages and disadvantages of both. Importantly, Vo et al. [285] spot the huge potential in integrating WSN into 5G, providing interesting point of views.

In line with the popularity of 5G networks, significant research on edge and cloud computing is continuously being conducted. According to Bolivar et al. [286], the scarcity of network resources at the edge is severe, despite the benefits brought by edge computing. Thus, the

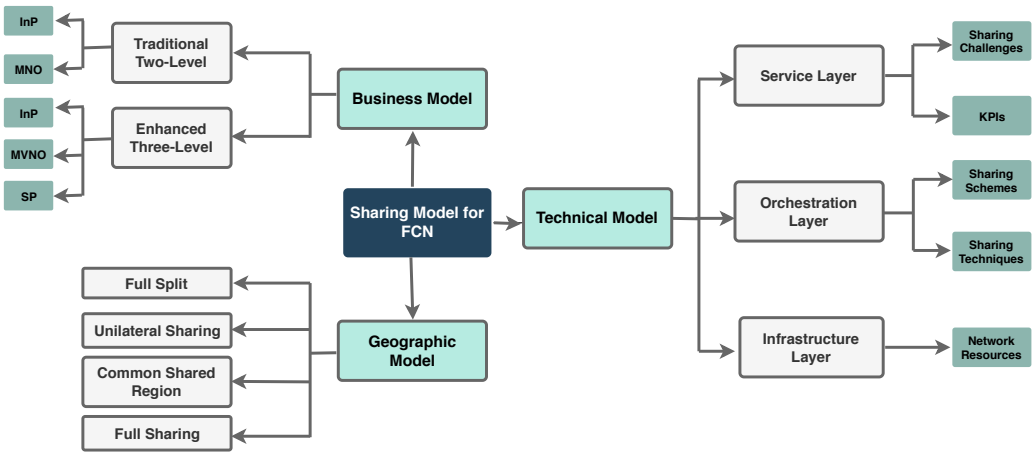


Figure A.6: Our comprehensive taxonomy for sharing of network resources in FCN.

amount of network resources is notably limited and efficient resource utilization is necessary. As demonstrated by these examples, virtualization techniques and SDN have gained incredible momentum in the past few years, and we expect this to continue steadily in the future.

A.3 Comprehensive Sharing Model for Future Communication Networks

Having examined resource sharing from a time evolution perspective, now we provide a classification based on research topics. This will answer the questions of what can be shared, how, and why. In this section, we propose a comprehensive taxonomy, summarized in Fig. A.6, which incorporates the commonly adopted general FCN architecture presented by Kliks et al. [18]. The taxonomy summarizes all the dimensions that have to be carefully designed and harmonized in order to create more efficient FCN. The first part of this section contains general overviews of sharing models for FCN, from technical, business, and geographic perspectives. The rest of the section further expands the resource sharing model from the point of view of infrastructure, orchestration, and service layers. Since a detailed review of the business and geographic models is beyond the scope of this survey, we provide only general information and point at the gaps which should be further addressed by research in these areas. The taxonomy presented in this section will help the readers identify what are the current gaps for sharing FCNs.

A.3.1 The Scope of our Contribution

Sharing of goods and means like heterogeneous network resources, goes beyond the technical tasks and assets. Questions such as: who to share the resources with, how to share, under which conditions, to what extent and where, require further attention before approaching

technical aspects of sharing. Following the early effort of Frisanco et al. [2] to explain the relevance of considering the business, and the geographic models altogether with the technical design, in Fig. A.6 we propose an extended taxonomy which portrays the flow of our survey. Following the guidelines to implement sharing in an existing network elaborated by [2], our comprehensive sharing model comprises three mutually coupled and heavily dependent parts: business, geographic, and technical models. Once these three components are selected, it is necessary to deploy the network assets in an optimal way. The first choice is to select which existing geographic sites will survive and which will be decommissioned. Concurrently, locations for new sites must be selected. Secondly, the existing equipment and technologies must be consolidated for sharing.

Business Model The business model describes the parties which are directly or indirectly involved in sharing, as well as the contractual relationships between them [2]. In a broader sense, it describes the rationale that governs and constrains the design of a technical sharing model. The sharing of heterogeneous resources is always enabled and performed by the technical model, but under the regulations, pre-defined rules, and criteria adopted by the corresponding business model. According to recent research [7], the resolution among network operators and their businesses, which agree on resource sharing based on virtualization techniques, can be obtained by two possible types of business models:

- *Two-level*: The traditional business model consists of the two entities: the MNO as a business entity which has subscribers but no infrastructure resources, and the InP as an entity with infrastructure resources but no subscribers. In such model, the virtualization tasks are assigned to InPs, which further manage those resources together with the MNO.
- *Three-level*: The enhanced business model consists of three entities: the MVNO, which now has the role of an intermediary between the InP and the SP; the SP which does not have enough infrastructure resources and thus has to lease and share resources from the InP's pool; and the InP.

Interestingly, Hultell et al. [238] envisioned that prospective business models should include network operators that can offer resources to specialized SPs many years before the development of above-presented three-level business model. Thus, another example of a potentially successful business model is the one which assigns the role of inter-connection provider to an arbitrary entity, which then supplies resources to SPs and MNOs [238]. Moreover, the idea to incorporate the Pay As You Go (PAYG) business model to enable sharing of resources that belong to IoT devices is presented by Kliem and Kao [265]. Such business model dictates resource pooling, which makes feasible the on-demand provisioning [265]. Concerning network slicing, in order to adequately address the requirements of managing different services and applications that are available within 5G network slices, Barakabitze et al. [269] list three possible business models for network slice commercialization: i) Business to Business (B2B), ii) Business to Consumers (B2C), and iii) Business to Business to Consumers (B2B2C). In the B2B model, resources are usually sold to enterprises by MNOs, while enterprises retain full control over their subscriptions. However, the B2C model allows customers to directly purchase resources upon their needs in an MNO-agnostic manner (i.e.,

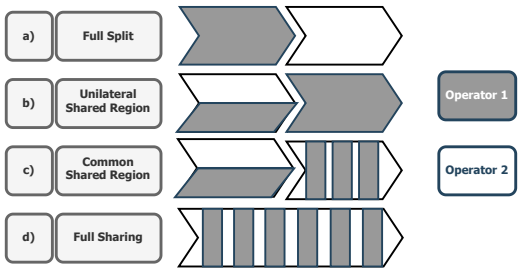


Figure A.7: The variants of the geographic model for infrastructure sharing [2]: a) Full split, b) Unilateral shared region, c) Common shared region, and d) Full sharing.

the provision of communication services is generalized, and users have a neutral attitude towards the MNOs). For this reasons, B2C poses significant challenges to overcome different requirements of different MNOs. Finally, B2B2C includes an intermediary between MNOs and customers, i.e., network slice broker, that allows different verticals to lease resources from InPs in a dynamic manner [269].

Furthermore, Akhtar et al. [228] observe the non-existence of corresponding business models as one of the potential reasons for failure to implement spectrum sharing, justifying the huge importance of business models in any sharing scenario. In this fashion, network operators or any sharing entity are not provided with sufficient incentive to share spectrum. Lastly, as a potential research direction, Rebato et al. [266] envision introducing innovative business models for resource sharing to better quantify a potential economic impact. Accordingly, it is clear that the development of adequate business models should keep the pace with technical models in order to increase performance gains (i.e., KPIs), anticipated from sharing resources.

Geographic Model According to Frisanco et al. [2], the geographic model describes each operator's physical footprint in a nutshell. In order to enable sharing, certain locations, operator's domains, and preferences based on the geographic position have to be known and established. In particular, for the infrastructure sharing, [2] and [249] gather and sum up the sharing scenarios with regard to the operators' geographic footprint. According to the area each operator covers in a multi-tenant scenario, a geographic model might include: standalone, full split, unilateral shared region, common shared region, and full sharing. Based on the studies in [2] and [249], we illustrate each of these cases for the simple scenario with two network operators tasked to provide coverage in a geographic area (Fig. A.7). Thus, Fig. A.7 shows two separate geographic areas which are covered either by one of the operators or by both of them. Respectively, the full split stands for only one operator, solely covering the whole area (Fig. A.7 a)). Within the case of unilateral shared region, despite the presence of both operators, the geographic territory is split between them following certain regulations and without sharing, but with opportunities to establish mutual service agreement (Fig. A.7 b)). Furthermore, if a certain operator has a full-coverage infrastructure and aims at leveraging it in order to gain additional revenues, then the unilateral sharing case would also apply. The small-scale operator is then allowed to enter the market without investing in infrastructure and suffering from risk related to small initial number of subscribers (e.g., large CapEx). On the other hand, if operators are of similar scale and, thus, want to operate jointly

in a certain area, they can approach sharing of their resources in a common shared region (Fig. A.7 c)). Finally, the full-sharing scenario is a theoretical base for deploying a technical model which can entail sharing heterogeneous resources in an end-to-end communication network [2], since it enables sharing of all resources between all network operators (Fig. A.7 d)). The goal of this brief review of business and geographic models is to emphasize that the paradigm of resource sharing combines several dimensions, whose common denominator needs to be identified. The performance of a shared network is deeply affected by other factors included in the business and geographic models. The overall choice upon any of the technical, business or geographic models limits the degrees of freedom for selection of the two remaining models [2].

Technical Model The taxonomy for the technical model can be seen in Fig. A.6 and its elements are discussed in detail throughout the Chapter. Since the main focus of this survey is on the technical aspects of network sharing, we dedicate the entire next section A.4 to it.

A.4 Technical Sharing Model

This section elaborates on the functional blocks drawn in the right hand side of Fig. A.6, which consists of the following three branches: 1. Infrastructure Layer, 2. Orchestration Layer, and 3. Service Layer. The detailed structure of our technical model is shown in Fig. A.8. First, the infrastructure layer consists of all shareable heterogeneous resources. Second, the orchestration layer consists of dedicated software platforms responsible not only for management, operation, and orchestration of heterogeneous resources in general as in [18], but also for sharing of those resources. Third, the service layer includes sharing challenges and KPIs, because the stakeholders, which are responsible for service management and delivery, must be aware of the benefits and the overall performance of resource sharing. Below, we first present the meticulous classification of network resources with respect to the network layers, followed by the most utilized sharing techniques, which are classified and described as the enablers of resource sharing. Finally, we point at widely used KPIs, which measure the success of the adopted and deployed sharing techniques.

A.4.1 Classification of Network Resources

In this section, we attempt to answer the question on which distributed and heterogeneous assets could and should be shared, in order increase utilization of such shareable assets. Based on the studied literature, we provide the classification of resources and present it in Table A.2. We categorize network resources with respect to the network layers introduced by Li et al. in [306]. The authors classify the network assets into four groups, depending on whether they belong to the physical layer, MAC, Internet Protocol (IP), or Virtual Private Network (VPN). Examples of classification criteria they use are isolation and customization among operators, efficient bandwidth utilization, etc. Rather than being requirements which must be fulfilled to enable sharing, we see these as the challenges, which we futher discuss

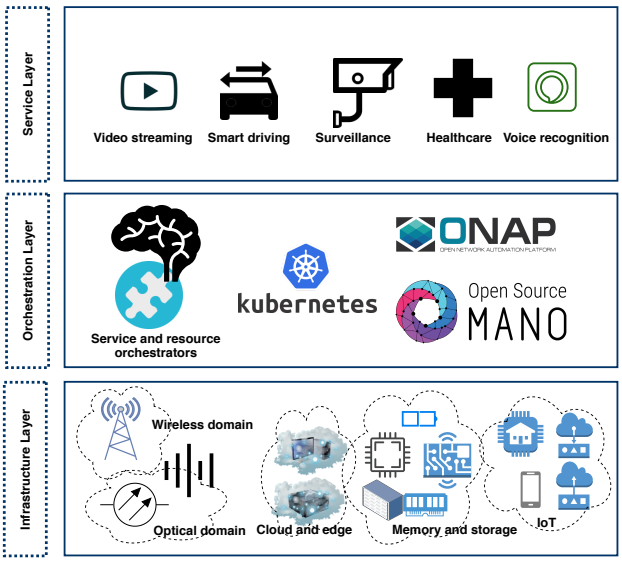


Figure A.8: The structure of our technical model.

in Section A.6. We collected all remaining types of resources which could not fit directly into any of the categories in a fourth group, labelled "Other".

Within each of these categories, we provide a further classification based on the nature of the shared resource, ranging from intangible (i.e., immaterial) resources, such as spectrum, to more tangible ones, such as network devices. From the perspective of Sarvanko et al. [242], immaterial resources are those that can be represented as abstract physical magnitudes. On the other hand, concrete resources are those such as real hardware, with processing capacity and ability to perform actions [242].

Physical layer resources The first group in Table A.2 classifies all physical layer resources. As one of the most pervasive, and yet the most influential asset, *spectrum* can be simply defined as a set of frequency bands with ability to enable electromagnetic signals to propagate.

The significance and the impact of radio spectrum is recognized by both researchers and network operators. For instance, in the work presented by Tridandapani and Mukherjee [234] sharing of spectrum is analyzed from the perspective of insufficient number of channels to orthogonalize all interconnecting network lines. In optical communications, most of the work on fiber spectrum sharing has focused on network redundancy, as link survivability is considered as one of the key concerns in network design, aiming to achieve fast service restorability against network failures [300]. When a failure such as fiber cut happens [289], back-up path should be available to restore the service. However, providing backup paths wastes resources, and thus they should be reused (i.e., shared) by services that can be preempted. In the context of EONs [290, 258, 259] Satkunarajah et al. [289] studied sharing of back-up resources in a pre-configured manner. This means that the backup paths are configured for sharing in advance. However, the multiple backup paths can share

Table A.2: Classification of Network Resources.

Network Resource Group	Type of Network Resource			Domain	Works
Physical Layer	Spectrum	General resources (channels)		All*	[234, 228, 225, 230, 229] [214, 210, 211, 209] [216, 212, 213, 215, 233] [236, 237, 243, 244, 241] [245, 246, 247, 248, 18] [287, 266, 288, 253, 258] [259, 263, 254, 268, 289] [290, 291, 292, 293, 294] [295, 296, 270, 268, 297] [298, 267, 15]
		Path protection resources		Optical	[299, 289, 290, 300, 258] [259, 260, 262, 224]
	Infrastructure	Transponders			[260]
		OEO regenerators	shareable		[262, 224]
			idle		[262, 224]
		Line cards			[260]
		Pure all-optical converters			[224, 261]
		OLT			[301]
		ONU			[226]
		ODN			
		Analog broadband repeaters			[302]
		Optical fiber			
		User interface	Sensors	IoT	[242, 303]
			Actuators		
		Sites		Wireless	[2, 304, 206, 249, 18] [16, 291, 294, 252]
		Towers			
		Air conditioning systems			
		Leased lines			
		Microwave links			
		BTS			
		RNC			
		SGSN			
		MSC			
		BNC			
		eNodeB			
	Energy			All*	[305, 303, 2]
	Connectivity (Air interface)				[242]
	Built-in	Memory	Storage capacity		[305, 303]
			Buffer space		
		Processing power			[242, 305, 264]
		Battery		IoT	
MAC Layer	Bandwidth			Optical	[306, 262]
IP Layer (VPN Level)	N/A				
Other	Network functions	Access		All*	[306, 262, 33]
		Transportation			
		Core			
	Functionality	Signal regeneration			
		Wavelength Conversion			
	Functionality extender	Localization engines		IoT	[303]
		Security accelerators			
	Social resources	Individual (user)		All*	[242]
		Group (community)			
	Computation burden			All*	[283, 264]

* All comprises wireless, optical, IoT, edge and cloud domains.

a given optical channel only if their corresponding primary routes are not expected to fail simultaneously [224] (i.e., they belong to the same Shared Risk Link (SRLG) group).

In the optical domain, statistical spectrum sharing can be implemented through the use of software-defined variable bandwidth transponders which can support variable data rates (base and peak), leading to variable bandwidth occupation. Khandaker et al. [258, 259] considered in their cost study the use of transponders and 3R regenerators capable of switching

between multiple rates, while Wang et al. extended the idea further, by considering shareable regenerators and line cards [260]. In particular, Wang et al. [260] proposed a method that enables optical transceivers to change bandwidth dynamically without service interruption. Their aim is to provide a mechanism for optical channels to match the statistical behaviour of network traffic, so that wavelength can vary dynamically between a base and peak rate. Their simulation study, based on well known optical topologies, show that statistical wavelength sharing can provide up to 200% gain in network capacity.

Two other shareable optical devices are presented by Ali [224] and Pedrola et al. [261], namely Optical-Electrical-Optical (OEO) regenerators and the pure all-optical converters. OEO regenerators are necessary for dealing with optical transmission impairments and/or realization of wavelength conversion [224]. In the implementation, one set of OEOs is shareable, while the another set is left idle [262]. The idle group is reserved for backup paths, as the other shareable devices are meant to be operational at any time. This type of resource sharing does not include multiple operators, and rather refers to sharing of costly components across multiple backup paths belonging to the same operator, enabling a reduction of capital costs for the operator. Nonetheless, sharing components increases the complexity of the network and control system, generating a trade-off between added complexity and reduction of total cost of network ownership, which needs to be studied.

Almost 10 years later, Pedrola et al. [261] tackled this issue of complexity increase due to sharing. They consider FCNs as networks that require high agility, for example implementing Sub-Wavelength Sharing (SWS), in order to cope with highly dynamic traffic patterns. They proposed an optical translucent network architecture, based on a mix of electrical regenerators and optical wavelength converters. The main issue is that sharing converters increases the complexity of the optical switches, which increases the number of optical gates required. The trade-off thus becomes one of relative costs between converters and optical gates. The outcome of their study highlight the conditions that need to be met in order for the sharing architecture to pay off: the cost of pure all-optical converters has to be at least two orders of magnitude higher than that of the optical gates, and similar or lower than the cost of 3R regenerators [261].

Moving towards optical access networks, Ruffini et al. [307] addressed sharing of optical devices in PONs. These networks are made up of: 1. Optical Line Terminals (OLTs), which are located in the Central Office (CO) of the InP, 2. Optical Network Units (ONUs), located at the user premises, and the 3. Optical Distribution Network (ODN) which consists of fiber cables and optical splitters deployed in the field. Their work, based on PON virtualization [308] enables multiple operators to independently schedule their capacity allocation. However, this creates a new issue, as virtual operators have no incentive to give away their unallocated capacity to their competitors. Thus they propose a novel mechanism [207] based on auctioning capacity between Virtual Network Operators (VNOs), thus restoring the sharing performance of PONs. In [309], they further extend their work to consider scenarios where the InP also operates as one of the VNOs, so that it cannot be considered a trusted third party. They thus re-formulate their auction as a distributed operation and demonstrate its feasibility on a blockchain implementation based on the Hyerledger Fabric.

In the wireless context, infrastructure sharing is mainly divided into passive and active [2, 18, 16]. Example of devices that can be shared passively are the RAN components, such as BTS and eNodeB. Site sharing is recognized as favorable from the perspective of

operators, due to the fact that lower overall number of occupied locations results not only in lower costs but also provides better environmental and aesthetic conditions [2]. Microwave links and leased lines, which usually form transmission networks between Base Station Controller (BSC) and BTS in 2G, and eNodeB and Radio Network Controller (RNC) in 3G and 4G, are considered as shareable and belong to the passive domain [2]. The RAN components such as BTS, BSC, RNC, and eNodeB can be shared actively as well. Multiple virtual radio access network instances are implemented by splitting the RAN elements into logically independent units running in one single physical device [2, 249]. In general, RAN virtualization supported by SDN provides isolation in terms of control plane functionalities for each sharing actor [269]. Due to the lack of practical SDN-based solutions for RAN sharing, Foukas et al. [310] developed a flexible and programmable SDN-supported RAN platform, i.e., SD-RAN FlexRAN. This platform offers southbound APIs for separating data and control planes, making the control plane programmable, technology-agnostic, and customizable to different sharing entities through the programmability of virtualized network functions. FlexRAN aims at facilitating resource sharing by exploiting the virtualization capabilities, which enable dynamic introduction of new MVNOs to the RAN, as well as on-demand customization of scheduling policies per each MVNO. Furthermore, Shantharama et al. [311] introduce LayBack, an SDN-based platform for extending sharing capabilities of RAN towards edge resource sharing. For the context specific to network slicing, additional SDN-based mechanisms for RAN virtualization are presented in [269, 312, 313, 314].

Besides the RAN, the core network can also be shared, but to a limited extent. This is restricted due to the confidentiality and performance requirement of the operator, because sharing the core network would imply sharing servers and network functionalities that are critical for running the network services [249]. These core network functionalities often contain confidential information pertaining the operator's business operation, and thus has to be kept within the operator's boundaries, which limits the level of shareability. Upon advances in NFV and SDN, and their inseparability from FCNs, sharing of core network resources has gained momentum. In particular, Meddour et al. pose an important design constraint for core network sharing: they propose the idea of FCNs with separated control and data planes through use of SDN. With such data and control plane separation, they state that core network elements such as Home Location Register (HLR), Gateway Mobile Switching Center (GMSC), and Gateway GPRS Support Node (GGSN) remain separate in one operator's core network, while at the same time Serving GPRS Support Node (SGSN), RNC, and Visitor Location Register (VLR) are available for sharing. This enables sharing the data plane of the core FCN but not the control plane, enabling service differentiation while maintaining confidentiality [249]. Likewise, but in LTE, both the control plane (MME and Home Subscriber Server (HSS)) and user plane (Serving Gateway (SGW) and Packet data network Gateway (PGW)) entities of the Evolved Packet Core (EPC) can now be developed as services on sophisticated GPP servers [314] that bring more flexibility in operation, enabling the opportunities for operator-specific requirements and customizations. The aforementioned is possible due to the fact that network operators can deploy multiple virtual instances of EPC at the same time, serving different categories of users [314] and sharing those resources with VMNOs or other sharing parties. In particular, such on-the-fly creation of virtualized core networks is enabled by virtualization technologies such as VMs and containers, and e.g., OpenStack as a platform for pooling of resources on demand [315].

From the IoT perspective, Pagani and Mikhaylov [303] consider WSNs composed of myriads

of nodes with highly heterogeneous characteristics, differing among each other in: structure and hardware components, processing and storage capabilities, communication interfaces, software applications, and available services. All these heterogeneous features provide a set of specific resources which belong to a certain IoT node (actuator, sensor, etc.). However, these resources are usually very limited, while devices are constantly being exposed to plethora of service requests [305]. Pagani and Mikhaylov also emphasize the importance of awareness of each other's resources and tasks among IoT nodes, in order to trigger mechanisms for sharing. Angelakis et al. [305] adopt similar approach by considering sharing in the context of splitting service requests into different interfaces with different resources. Furthermore, energy as a shareable resource among IoT devices is examined by Kouvelas et al. [283] in their theoretical graph theory-based sharing algorithm. This specific algorithm is created under the assumption that excess energy transmission between microgrid interconnected IoT devices is feasible.

MAC layer resources Li et al. [301] introduce slice and frame schedulers to enable bandwidth sharing in XG-PONs. In this context, slice scheduler decides on the slice owner for each frame, while the frame scheduler enables the operator to schedule the bandwidth resources of the frame for its subscribers with customized bandwidth allocation schemes [301]. However, isolation and customization problems have arisen in such scenario, for which a novel solution based on intra-frame sharing was developed in [275], as discussed later in the paper.

Other resources As the last but certainly not least, we refer to other resources that could not fit into designated network layers. First, we briefly turn to the *social resources* mentioned, for instance, by Sarvanko et al. [242]. Although this type of resources is beyond our scope, social resources can be perceived as an integral part of the users or users' perception, since they are important for the cognitive and cooperative sphere of sharing. Sarvanko et al. underline the importance of users' decisions on what, when, and with whom to share, in alignment with the corresponding KPIs which are the outcome of such sharing process. Meddour et al. [249] also refer to this type of resources, but in the form of Radio Frequency (RF) engineering support in the sharing resources chain.

We further present some explicative attempts to share the *functionalities* among different sharing entities in IoT, and optical networks. The IoT devices can share not only excess energy, as presented by Kouvelas et al. [283], but also functionalities. One example is presented by Silva et al. [264], in which cellular unconstrained and IoT constrained devices share resources and functionalities. Thus, the benefits are mutual, because unconstrained devices can assist constrained ones during the service operation by proper task offloading [264]. Furthermore, such offloading of computation-intensive tasks from the resource constrained devices to the cloud environment is recognized as a beneficial and promising solution for FCNs in general. It is supported by MEC, which is one of the key technology pillars for 5G networks [33]. As stated by Taleb et al. [33], MEC provides a shared pool of resources, which can be scaled dynamically. Interestingly, sharing of functionalities is studied in the optical domain as well. For example, Manolova et al. [262] propose the use of regenerators both for signal regeneration and wavelength conversion, providing additional flexibility to their resource allocation algorithm.

An important elaboration of network functions as a resource that can be shared is given by Taleb et al. [33]. In the context of MEC, VNFs deployed in the form of virtual machines and containers can be dynamically allocated and re-allocated, and thus shared. Since traditional access, transportation, and core network functions can be transformed into virtual network functions, we list some of the general mechanisms to share VNFs. 5G-Transformer project[3] aims to transform today's mobile transport network into an SDN/NFV-based platform that manages slices tailored to the specific needs of vertical industries, by customizing VNFs. This project recognizes the potential in developing new mechanisms for sharing VNFs by multiple tenants and slices. As VNFs are today base components of network services, it is not unusual that they are common for various network services in parallel. Therefore, Malandrino et al. [316] study the opportunities of VNF sharing by considering multiple criteria, such as: i) conditions upon which VNFs can be shared, ii) distribution of the workload per virtual machines that run shared VNFs, and iii) possibilities to prioritize service traffic within shared VNFs. Thus, authors propose FlexShare optimization algorithm for VNF sharing, and show that this algorithm outperforms baseline solutions in terms of achieved KPIs such as service deployment cost, and total delay [316].

At this point, all of the resources that we recognized as shareable in the considered literature scope have been introduced. The rest of the section is dedicated to illustrating their sharing potential and how can these be exploited to achieve target KPIs.

A.4.2 Sharing Techniques

In this section we discuss some of the most frequent techniques (Table A.4), used to pool and share network resources.

It is important to acknowledge that a large variety of available network resources, such as different technologies and services in FCNs, and in particular in 5G network, bring huge heterogeneity to the network. To achieve the promised connectivity, new services and applications, and the benefits of full capacity in 5G networks, the users need the ability to access infrastructure deployed by different operators, not only the one for which they have a subscription. Multi-tenancy plays a key role to enable such scenario.

Virtualization In order to keep up with the agility required to deliver the 5G KPIs across heterogeneous networks, 5G networks introduce virtualization and softwarization [14, 15]. Although the definition of *virtualization* depends on the application domain, a quite general and straightforward rationale is provided by Van De Belt in [5]. Van de Belt et al. interpret it as a technique which enables network services to observe and use network resources in a manner which is independent from the underlying physical infrastructure. Importantly, a likely outcome of this ability is the possibility to use these resources in a scalable and customizable way. The utilization of resources can be aligned with the service requirements, gaining significant reduction in time and resources for network deployment and operation [16]. Regardless of the domain it applies to, virtualization can be comprehended as abstraction, isolation, and sharing of heterogeneous resources among multiple actors (network

[3] 5G Transformer: `http://5g-transformer.eu/`

Table A.4: Classification of Sharing Techniques.

Sharing technique		**Domain**	**Works**
Virtualization		All*	[270, 7, 277, 226, 6] [11, 256, 284, 266, 221] [317, 271, 318, 14, 294] [319, 16, 15, 18, 287] [320, 269, 272, 321, 315] [322, 323, 324, 312, 314]
Software defined networking			[277, 226, 7, 317, 11] [256, 319, 221, 271, 14] [16, 15, 18, 287, 320] [318, 269, 272, 321, 315] [322, 323, 324, 312, 314]
Resource brokering			[11, 7, 325, 6, 18, 319]
Network slicing			[6, 271, 270, 319, 269] [272, 321, 315, 322, 323] [324, 312, 314, 326, 327, 328]
WSN management middleware		IoT	[250, 284]
On-demand provisioning			[265]
Resource pooling			
Registration and resource provision accounting	ranking		[264]
	earning credits		
Assigning services to interfaces with heterogeneous resources			[305]

* All comprises wireless, optical, IoT, edge and cloud domains.

operators or users) in both wireless [6, 7] and optical domain [11, 12], achieving a certain degree of isolation between all sharing units [7].

For the wireless domain, Zhang [7] emphasizes one important characteristic of virtualization, which is the capability to approach abstraction and isolation of physical layer resources, and to map them into specific virtual networks. Zhang's consideration of virtualization as an umbrella which covers several different realms designated according to the part they play in the end-to-end FCN is presented in [7]. Based on his approach as well as Liang's and Yu's work [16], one can notice that the RAN as well as the core network can be virtualized completely or up to a certain level. Indeed, sharing of wireless access and infrastructure have become easier to achieve after the development of virtualization techniques. Regarding core network sharing, various sources [329, 330, 277, 331] propose techniques for virtualization of EPC in a mobile network, as well as corresponding SDN-based control architectures. Such control architectures are capable of dynamic reconfiguration of the transport network in order to reroute the traffic dynamically to the closest available virtualized EPC [277]. For instance, the virtualization techniques presented by Costa-Requena et al. in [329] and [330], have proved their beneficial nature, since they provide better utilization of resources and cost reduction of 7.7%.

Furthermore, many authors indicate that isolation among resources represents a crucial part in the virtualization process [306, 301, 16, 265, 256, 7, 271], since it directly impacts the sharing. Thus, the isolation has to be studied with a more prominent attention and as an essential challenge. Moreover, Li briefly explains the difference between applying virtualization techniques in optical and wireless domains, and illustrates necessary modifications which have to be made in wireless networks in order to make virtualization functional [332]. He also anticipated that the SDR is a valuable asset which can further enhance the performance

of virtualization techniques.

Some of the advantages enabled by virtualization are flexible and dynamic management of resources that can enable network operators to provide new types of services [16]. Such flexibility could not be possible without certain set of previously inaccessible virtualized resources. Furthermore, Van De Belt et al. accentuate improved security and protection when virtual networks are deployed on top of the existing infrastructure thanks to the inherent isolation of network resources [5]. Another important improvement brought by virtualization techniques is examined by Afraz et al. [226]. Since these techniques provide dynamic control and management [16], which the network operators can further align to the users' requirements, the concept of multi-tenancy would be more acceptable and trustworthy solution than ever before. Empowered by SDN, network programmability and control plane centralization, virtualization of network resources, and functions can facilitate multi-tenant scenarios by providing the VNOs with immediate access to network functions without any intervention from the InP [226, 6]. Afraz et al. focused on optical domain and concluded that virtualization of devices such as ONU and OLT can make the PON significantly flexible.

From a business perspective, Chowdhury and Boutaba in [333] envision network virtualization as the decoupler of the traditional Internet Service Providers (ISPs) business model into two separate and independent entities, namely the InPs and the SPs. The specific roles of these two entities are to manage physical infrastructure and to create virtual networks by aggregating various resources from different InPs, respectively.

Recently, [221, 284] studied how virtualization can be applied in the IoT ecosystem. Due to the fact that IoT networks suffer from resource constraints, virtualization seems to provide many opportunities in their deployment and operation. In their survey on virtualization techniques in the context of IoT resource management, Zahoor and Mir in [221] see virtualization as the approach that can play an important role in maximizing resource utilization and managing the resources. Yildirim and Tatar [284] propose Node-based Virtualization (NoBV) and Network-based Virtualization (NeBV) as a way to apply virtualization into WSNs, and these specific use cases are further elaborated in the Section A.5. In the latter part of this subsection we refer to other techniques listed in the Table A.4.

SDN According to plentiful of sources, SDN can be defined as an emerging programmable architecture which decouples network control from data (sometimes also referred as forwarding) plane. However, the seminal point for such control and data plane separation lays in the need for effective and dynamic resource and processing power management in modern computing environments [18]. Zhang [7] concisely elaborates the key features of an SDN architecture, explaining that the centralized control in 5G networks supports and enables service-oriented operation, which is dynamic, easily manageable, cost-effective, and customizable to the emerging and 5G-specific applications (i.e., eMBB, mMTC and uRLLC). An example of control plane implementation is presented by Raza et al. [277], within their approach of dynamic resource sharing for C-RAN with optical transport network. This approach takes advantage of a hierarchical SDN controller as a global orchestrator which harmonizes transport resources, in line with the spatial and temporal variations of the wireless traffic. Focusing on the wireless access, Rebato et al. [295] emphasize its sharing opportunity through joint utilization of SDN and NFV, as a viable option to leverage macro-diversity

in mmWave bands.

Another perspective of applying SDN in 5G heterogeneous networks is presented by Akhtar et al. in [228]. Their approach embraces principles of centralized management with hierarchical control domain in order to globally control the entire network despite distributed inputs arriving from users. The centralized approach inevitably raises concerns on scalability and latency, but the authors address them by balancing the task distribution among the controller and the BTSs. This can be achieved by limiting the controller to manage only global network rules in the back-end, while the BTSs form the front-end interacts with and manages the user devices [228].

Network slicing and Resource brokering Crippa et al. [317] introduced the project 5G NOvel Radio Multiservice adaptive network Architecture (5G NORMA) and its network-of-functions-based architecture suitable for supporting a wide variety of services with various requirements. This architecture is one of the first applications of novel concepts such as *network slicing* and multi-tenancy [317]. According to ETSI's NFV MANO [334], network slice is defined as a set of network functions and resources which are necessary to run these functions, forming a complete logical network capable to meet the network characteristics required by end-to-end services. Thus, network slices are nothing else than logical virtual networks based on the physical shared infrastructure, allocated and customized according to the QoS demands [270, 7, 271].

Network slicing as a technique for enabling resource sharing among multiple tenants is considered a key functionality of next generation mobile networks [322]. Cabellero et al. [322] provide an illustrative practical exemplification of creating network slices, explaining that each slice consists of VNFs that jointly form the network services that run on top of heterogeneous infrastructure. According to Caballero et al., the deployment of network slicing starts with a slice creation phase (i.e., an end user requests a slice from the NS catalogue and tenants responds with slice instantiation), and continues with a runtime phase (i.e., triggering operation of functional blocks allocated within slices).

The concept of network slicing is gaining significant attention from the telecommunication industry, with an accent on providing network as a service for different use cases [271]. Khan et al. [291] present the core modules that enable dynamic allocation of RAN network slices with dedicated spectrum and resource scheduling functions. Their results show benefits and trade-offs of spectrum sharing between RAN tenants. Similarly, based on the 3GPP's DÉCOR technology, Kiess et al. [331] investigate methods to upgrade existing heterogeneous networks with a slicing mechanism that requires minimal changes to select and configure the slices. In the scope of resource sharing, the SDN controller plays the role of either an *orchestrator* or a *resource broker*. The actual role depends on the architectural designer's preference. As an example, Samdanis et al. [6] introduce the on-demand capacity broker, whose role is to facilitate on-the-fly resource allocation. In this paper, the authors provide a detailed overview of the new control architecture installed on the top of existing 3GPP networks with a network slice broker as brain. Their approach is similar to those presented in [228, 262, 7], since they also adopt a hierarchical control architecture.

The compound of stringent QoS requirements for advanced 5G services and applications, and dynamic wireless environment poses a significant challenge to existing management tech-

niques [326]. Therefore, Isolani et al. raise the importance of performing slice orchestration and IEEE 802.11 MAC management at runtime for the end-to-end QoS [328, 326, 327]. In [328] they propose an algorithm for on-the-fly end-to-end slice orchestration and IEEE 802.11 MAC management based on the application's QoS requirements. The main purpose of this algorithm is to periodically re-calculate and adjust the resources allocated to each network slice based on the current QoS demand. In their realistic experimentation within the testbed environment consisting of one centralized SDN-based controller, one Access Point (AP), and two clients, Isolani et al. [328] show that their algorithm brings significant improvements in QoS, i.e., throughput, latency, and reliability. Furthermore, in [326] Isolani et al. go further and exploit the flexibility of slice airtime allocation considering both resource availability and stringent latency requirements for uRLLC, towards achieving the optimal allocation of network slices in IEEE 802.11 RANs. To assign different airtime configurations per network slice, Isolani et al. [326] use a scheduling policy, enabling prioritization among slices. As expected, the optimal allocation of slices depends on the number of slices to be allocated, and the strictness of QoS requirements for each of the slices [326]. In order to improve the allocation of slices in an SDN-enabled 5G network infrastructure, Isolani et al. [327] have recently upgraded their SDN-based management framework by gathering fine-grained end-to-end network statistics via advanced monitoring techniques - Inband Network Telemetry (INT), that enable higher level of granularity in monitoring dynamics of wireless environments. Given the monitoring reports, the slice orchestrator performs slice re-arrangement to meet QoS requirements, and SDN management entity distributes the flows to the isolated slices.

Considering network slicing from a federation perspective that includes multiple administrative domains, Taleb et al. [324] develop a federated management architecture with multi-domain Service Conductor plane that consists of: i) service broker, which performs the admission control and negotiation once a tenant requests slice, and ii) service conductor, which analyzes successful requests forwarded by service broker, and selects corresponding domains before instantiating a cross-domain slice coordinator for an allocated network slice instance [324].

As every solution comes at a price, the concept of network slicing is not an exception. A likely issue in network slicing for virtualized FCN is a potential underutilization of network resources, which, for example, can occur during network congestion [318]. In order to cope with this challenge, Gang and Friderikos [318] propose optimal and near-optimal inter-slice sharing between tenants. For instance, Vlachos et al. [319] reinforce sharing models that result in better resource utilization, with a specific focus on so-called cross-slice coordinator, which is presented as an extension to the SDN/NFV framework.

On-demand provisioning and Resource pooling Given the enormous increase in number of devices, the users' IoT environment will suffer from scalability issues. One of the attempts to address issues directly caused by the IoT proliferation, is presented in [265] by Kliem and Kao. To resolve the resource management issues, they map the concepts which are characteristic to the cloud computing domain like on-demand provisioning, elasticity, and resource pooling onto the IoT ecosystem.

Assigning services to different interfaces Due to the fact that almost every IoT device is equipped with numerous interfaces, Angelakis et al. [305] tackled the problem of *assigning different services to different interfaces*, in order to customize heterogeneous resources to the services requirements.

A.4.3 Key Performance Indicators

As Table A.5 indicates, we recognize numerous KPIs widely used in the research community. These indicators are used to evaluate and compare the performance of proposed and existing use cases, algorithms, or architectures.

The summary in Table A.5, shows that the authors mostly use CapEx and OpEx to emphasize *cost efficiency*. According to [207], there is an assumption that network sharing can provide the required economic incentives if properly implemented. In the wireless domain, Oliva et al. [136], within the scope of the 5G Transformer project, state that infrastructure sharing among tenants, based on the network slicing, is supposed to reduce OpEx. Furthermore, in the optical domain, Afraz et al. in [226] convey the statement from the Broadband Forum (BBF) standardization body[1], in which sharing of network infrastructure is a preferred means to reduce network costs and to make network scalable.

Costs play a very important role for any market player such as MNO, InP, MVNO, SP, end users etc. in the business model of a communication network. However, other KPIs, such as QoS parameters and spectral efficiency are also widely exploited to evaluate sharing of network resources from technical perspective. Since *spectrum* is a highly limited and precious resource, it is not surprising that many publications tackle spectral efficiency as a KPI. The remainder of the section shortly presents how the authors incorporated different KPIs into their specific use cases, algorithms, or architectures in order to evaluate their performance.

Wireless domain-related KPIs Since the idea of FCN is created to support the three generic classes of services, namely mMTC, uRLLC, and eMBB [343, 4], it is important to understand how resource sharing affects their *QoS* parameters. The services falling into these three categories most importantly differ with respect to required latency, number of connected devices, and throughput. In the context of throughput requirements in 5G networks, an interesting approach presented by Khan et al. in [291] facilitates specific radio resource segmentation and management through distinct slice-specific MAC procedures to enable granular spectrum sharing. Their results confirm that achieving more granular spectrum sharing ultimately leads to increased throughput.

Bousia et al. [206] justify significant improvements in the *network energy efficiency*, and QoS for MNOs which share infrastructure according to their proposed algorithm. Adopting a game theory approach, their algorithm facilitates switching off of the redundant BTSs while achieving high reduction in the total expenses. Due to the probabilistic nature of arrivals of service requests, switching off of the BTSs can increase probability of a service request being blocked. Therefore, for such systems the case of any general service requests not being successfully established in the network becomes the most important KPI. In [344] Bluemm

[1] https://www.broadband-forum.org/

Table A.5: Classification of Key Performance Indicators (KPIs).

Key Performance Indicators	**Domain**	**Works**
Blocking probability/Blocking rate	All*	[263, 335, 336, 266, 262, 257] [299, 258, 259, 289, 290]
Capacity Gain		[260, 325, 331, 337, 317, 338] [287, 18, 207, 306, 234, 227]
Quality of transmission		[262]
Quality of data		[250]
Resource utilization ratio		[299, 317, 266, 338, 287] [228, 18, 277, 262, 256, 299]
Control plane scalability		[256]
CapEx		[6, 335, 206, 249, 302] [250, 7, 325, 339, 331] [337, 266, 338, 18, 207, 224] [261, 277, 306, 226, 301] [262, 256, 260, 283, 305, 284, 268]
OpEx		[6, 335, 206, 249, 302] [250, 7, 325, 339, 331, 337] [266, 338, 296, 18, 207] [224, 261, 277, 306, 226, 301] [262, 256, 260, 283, 305, 284, 268]
Coverage area		[249, 238, 266, 268]
Data rate		[336, 238, 339, 253, 337, 266, 338, 228, 207]
Network energy efficiency		[206, 261, 283]
Spectral efficiency		[340, 290, 300, 268, 214] [211, 209, 216, 212] [213, 215, 288, 233, 236, 237] [243, 244, 248, 245, 246] [247, 241, 253, 258, 263] [254, 291, 293, 295, 296] [297, 267, 15, 259, 341, 321, 342]
Quality of service		[325, 331, 337, 317, 338] [287, 18, 207, 306, 234] [227, 284, 268, 291]
Quality of experience		[335, 253, 337, 317, 340, 287]
Probability of achieving peak rate		[258, 259]
Duration of investment payback period		[6, 339, 268]
Mobility		[268]
Complexity of site acquisition	Wireless	[6, 339, 268]
Survivability	Optical	[262, 289, 300]
Regenerator availability		[262]
Average regenerator usage		[262]

* All comprises wireless, optical, IoT, edge and cloud domains.

et al., demonstrated a similar concept on a testbed prototype, where an SDN controller could selectively put into sleep mode Baseband Unit (BBU) and Remote Radio Head (RRH) of an SDR-based C-RAN.

Farhat et al. [335] investigated resource sharing in a multi-operator 5G network, where VNOs agreed on the percentage of the resources shared with guest users. The incentive for sharing in this case is the increased user satisfaction due to lower blocking rates. Their simulations

point to an additional advantage in higher profits as the operators share more capacity, although they also recognize a trade-off between users' and operators' satisfaction (i.e., higher revenue/lower expenses) [335]. Another example which corroborates the benefits of spectrum sharing is presented by Hultell et al. [238], showing the scenario in which two or more license holders cooperate and share frequency carriers. Besides improved spectral efficiency, sharing also provides higher data rates with wide-area coverage.

An interesting evaluation of case studies for cost savings across different user density scenarios, is presented by Meddour et al. [249]. They based their evaluation on comparing different infrastructure sharing models, such as Multi-Operator Core Network (MOCN) and Gateway Core Network (GWCN), presented in Section A.5, including model sub-types based on whether they include backhaul or spectrum sharing. The importance of their contribution lays in the conclusion that *the highest savings in CapEx and OpEx are provided by the GWCN implementation*, since it allows maximum degree of sharing between the operators [249]. With similar use cases, Samdanis et al. [6] inspect 3GPP sharing principles and mechanisms for FCNs with multi-tenancy. They argue that in urban areas, sharing can greatly simplify the complex and long processes of site acquisition due to spectrum regulation limitations. Similarly, sharing can reduce the *network investment payback period* in rural areas [6].

Adding to the arguments in favor of infrastructure sharing, Nokia estimated that 20-30% cost savings from site sharing, and 30-40% cost savings from sharing both sites and RAN can be achieved [302]. Likewise, but from the perspective of Enhanced Cloud RAN (EC-RAN), Yu et al. [296] provide illustrative results showing that resource sharing between cloudlets can significantly improve the performance of 5G-enabled vehicular networks, and reduce system operation cost.

Authors usually approach the resource sharing problem by creating a suitable use case, sharing algorithm, or architecture, and testing its performance in terms of sharing benefits against a choice of different scenarios (i.e, varying their input simulation parameters). Except for two notable examples, we will not go into details of many such approaches and KPIs used therein, since they can be easily found within the taxonomy provided in Table A.5. In the first example, Kibilda and DaSilva [252] introduced an innovative regime for infrastructure sharing — so-called Networks without Borders, which aims at efficient provisioning of *coverage* among all involved operators. Their idea in the background of regime's operation is to dynamically select a wireless network which: i) represents the most suitable choice for the upcoming user service request, and ii) provides the lowest possible cost for an operator [252]. Similarly, Cano et al. [339] utilize Mixed Integer Linear Programming (MILP) to find the most suitable solution for resource sharing among the network operators, given as input techno-economic parameters, such as throughput for the end users, Return of Investments (ROI), pricing models, etc.. Their output is expressed in terms of most suitable solution for sharing resources among operators. Numerous publications reflect the huge interest in sharing network resources in mmWave bands, and some of them also point at their essential advantages in terms of KPIs. For instance, Rebato et al. [266, 295] studied the potential of mmWave spectrum and infrastructure sharing by assessing the achieved capacity gain. They point at two major benefits of sharing: i) super-linear increase in user rate with increase in cell density due to signal being power-limited, ii) decrease in *blocking probability* [266, 295]. They also present how to cope with increased interference in mmWave bands when it comes

to sharing.

Georgakopoulos et al. [338] and Kostopoulos et al. [287] agree that energy and *resource utilization efficiency* are key factors for sustainability in 5G networks. To support the previous statement, Georgakopoulos et al. conducted simulations that resulted in significant energy gains in comparison to the scenarios without sharing. From the perspective of the COHERENT project, Kostopoulos et al. developed a programmable 5G control plane, which pointed at huge opportunities in efficient control of network resources in the form of programmable 5G control framework, mostly because of increased capacity, spectrum and energy efficiency, as well as *QoE* that can be achieved. In their multi-operator resource allocation scheme, Marzouk et al. [342] studied static and adaptive spectrum sharing among MVNOs, by providing them with fair distribution of resources, and adaptive amount depending on their bandwidth requirements, respectively. Similarly to Marzouk et al. [342], Gang, Frederikos et al. [321] introduce tight and loose coupling, based on whether shareable resources are predefined or dynamically allocated. Although based on theoretical assumptions, Marzouk et al. [342] present an interesting way on studying how different distribution of shared resources can affect spectrum utilization efficiency, average throughput, and users' satisfaction. Through their simulation results, Marzouk et al. [342] show that adaptive sharing utilizes spectrum more efficiently in case of low density of users. Such approach might be interesting to test in the case of network slicing, where potential underutilization of resources in specific slices might occur.

Optical domain-related KPIs Regarding the previously discussed resource utilization efficiency, Zhang et al. [341] investigated how to reuse idle fiber spectrum. Their simulations emphasize that resource utilization efficiency can be improved to a greater extent if the interference is reduced in optical networks that adopts flexible bandwidth allocation. As already mentioned, installation and operation of devices such as optical transponder cause significant cost to the network operators. In this respect, Raza et al. [277] proposed and tested a novel strategy that resulted in up to 31.4% of cost savings from decreasing the number of optical transponders through dynamic sharing. They also mentioned that this would increase even further with 5G networks, due to the use of high-density by small cells [277]. Cost-effectiveness can be achieved not only by sharing optical devices but by sharing network functionalities as well. As previously mentioned, Manolova et al. [262] used this approach with the specific objective to ensure requested Quality of Transmission (QoT) and backup resources for improved survivability. Several KPIs are tightly coupled with efficient use of backup resources, which are typically required to provide high level of resilience in optical networks, but pose a trade-off between level of availability and efficiency in spectrum utilization. Similarly, Ning-Hai Bao et al. [299] and Chen et al. [290] evaluate sharing of backup resources in order to achieve higher spectral efficiency, and to decrease the probability of blocking service requests.

Blocking probability is widely used to evaluate the performance of network optimization algorithms, such as routing and wavelength assignment, in optical networks [257, 11]. These references utilize blocking rate of service requests in virtualized EONs to experimentally prove the performance of their sharing framework. Furthermore, an essential and yet quite general question has arisen from the study of isolation among virtualized optical networks provided by DeLeenheer et al. [256]. This work tackles the importance of trade-off between network

resources utilization and control plane scalability and proposes a resource sharing algorithm that reduces the number of wavalength channels required by 10%. Due to its significance we will discuss this work in Section A.6, as an implementation challenge.

With regard to the network slicing, Crippa et al. [317] provide a detailed architecture of network slicing management framework. They propose the use of a controller for each slice, which is responsible for preparing the resources for a given slice and to manage those resources. These controllers set the input values according to the specific service QoS/QoE requirements and constraints.

IoT-related KPIs Although IoT devices are typically low cost, they are deployed in large number, thus it is important to consider all their operational costs (e.g., including energy consumption). Yildirim and Tatar [284] state that resource sharing between heterogeneous WSNs leads to significant cost savings and reduction in latency, in particular for large IoT systems such as smart cities. Likewise, Kouvelas et al. [283] facilitate micro-grid within IoT systems for the sake of sharing energy locally and reducing the overall costs. In their already mentioned work, in which they assign different service requests to interfaces with heterogeneous resources, Angelakis et al. [305] also strive to meet QoS requirements and to minimize costs. Accordingly, their numerical cost analysis considers both costs of activation of services' splitting, and their distribution among interfaces. Their MILP-based algorithm demonstrates the impact of the total number of algorithm iterations, focusing on the trade-off between the minimum number of iterations and minimum cost. Looking back at the approach presented by Yildirim and Tatar in [284], time savings in time-critical IoT systems are achievable if the client evaluation entities (i.e., command/queries, data aggregation, and data fusion algorithms, etc.) are brought closer to the MBSS because the time needed to notify that resources will be shared is ultimately shorter [284].

In wireless networks, the time-variant nature of the transmission medium can strongly affect IoT applications and their strict QoS requirements [227]. Thus, similarly to the sharing architectures presented by Kunst [325] and Crippa et al. [317], Shi et al. [227] provide an IoT architecture with two-layer information base. The user level is indicated as a resource management level, which tracks QoS as well as QoE, and according to the predefined threshold coordinates new service requests seeking for new and more reliable routes. The network level instead reconfigures the networking resources to overcome the negative effects caused by changes in network states. Since the resources in IoT networks are shared by all users, the resource requests from one user might affect the network state, and the network level thus either performs resource adjustments limited to network, or provides a dynamic share or rent of frequency from other networks [227].

To sum up the section, we briefly mention spectral efficiency approaches related to IoT. For instance, Zhang et al. [212] claim that advanced spectrum sharing schemes such as CR, NOMA, D2D, IBFD, and LTE-U improve spectral efficiency for IoT applications. Other approaches include the use of unlicensed mmWave band. The authors also suggest new research directions in investigating the integration of multiple spectrum sharing techniques to address the highly heterogeneous nature of 5G networks. Finally, there are several observed and yet very important challenges related to LTE/NR UL sharing which is expected to benefit IoT applications [268]. In particular, this approach generates trade-offs between

Table A.6: Classification of Decentralized Sharing Models - Infrastructure Sharing.

<table>
<tr><th colspan="4">Type</th><th>Description</th><th>Domain</th><th>Works</th></tr>
<tr><td rowspan="3">RAN-only Sharing</td><td colspan="3">Passive RAN Sharing</td><td rowspan="5">Wireless</td><td rowspan="3">[206, 7, 226, 2, 18]
[266, 295, 287, 249, 310, 340]</td></tr>
<tr><td rowspan="3">Active RAN Sharing</td><td>MORAN</td><td>no core network sharing</td></tr>
<tr><td>MOCN</td><td>MORAN + frequency pooling</td></tr>
<tr><td>RAN + Core Network</td><td>GWCN</td><td>MORAN + core network</td></tr>
<tr><td rowspan="3">RF power distribution</td><td rowspan="2">Common DAS</td><td colspan="2">Sharing Analog Broadband Radio Repeaters</td><td rowspan="3">[302]</td></tr>
<tr><td colspan="2">Optical Fibers</td></tr>
<tr><td colspan="3">Sharing RF Power Among Operators via DAS</td></tr>
<tr><td colspan="5">Sharing OEO Tables</td><td>Optical</td><td>[224]</td></tr>
<tr><td colspan="5">IoT-related</td><td>IoT</td><td>[264, 283, 227, 284, 303]</td></tr>
</table>

spectrum availability and coverage, spectral efficiency and Downlink (DL)/UL coverage balance, transmission efficiency and latency, and seamless coverage and deployment investment [268]. Due to their relevance for incorporating IoT into FCNs, they are elaborated within Section A.6.

A.5 Use Cases

In this section we aim to present practical use cases, taken from the literature, that exemplify the sharing of heterogeneous and distributed resources. Here, the term use case refers to the specific model for resource sharing, which assigns roles to the participants and specifies steps in the sharing procedures. Such participants then follow these procedures to improve their KPIs. Tables A.6, A.7, and A.8 summarize the studied use cases.

While inspecting the features of various use cases presented by different authors, we noticed that sharing models primarily differ among each other in the way the control and management entities are organized and implemented. Accordingly, we group them into two categories: *decentralized/distributed* and *centralized*, which are presented in Tables A.6, A.7, and A.8. Furthermore, we evaluated both categories from the perspective of infrastructure and spectrum sharing.

The reason we apply this differentiation between sharing models, is to better suit the typical organizational structure of FCNs' control planes. Due to the synergy of SDN and NFV, FCNs' control planes can be organized in a centralized, hierarchical, and distributed manner [7]. In the first case, the whole control entity is made of only one SDN controller having a global view of the whole network, which makes it easier to implement but hard to scale. The distributed case, however, reflects the spread out nature of the control entity, consisting of several SDN controllers which communicate among each other to increase their local knowledge [7]. This distributed control plane architecture is suitable for stringent 5G service requirements, especially because of the reduced latency, but at the same time it is very hard

Table A.7: Classification of Decentralized Sharing Models - Spectrum Sharing.

Type				Description	Domain	Works
General	LAA			standardized version of LTE-U	Wireless	[212, 291, 214]
	LTE-U			coexistence between LTE and WiFi users on the same WiFi 5GHz channel		
	LTE-WiFi Aggregation			LTE signal uses WLAN connections to increase capacity		
	Multefire			operates only in unlicensed band and combines LTE performance with WiFi simplicity of deployment		
	Cognitive Radio	Overlay				
		Underlay				
	D2D			direct communication between two nodes when BTS is far away		
	IBFD			signal transmission and reception at the same time on the same frequency band enabled		
	NOMA			BTS allows connection on the same spectrum band to multiple users		
	Statistical Spectrum Sharing			switching between basic and peak rate	All*	[207, 258, 260]
	Sharing Tables					[224]
	Cooperative Spectrum Sharing	CSA		fixed number of spectrum slots per connection		[263, 262]
		Expansion	DAD	spectrum sharing allowed between neighboring connections		
			ACN	- spectrum re-allocation not allowed - consumption of resources from connections with potentially more available resources		
		Re-Allocation	Shift ACN	spectrum re-allocation allowed with restrictions		
			Float ACN	no restrictions on spectrum re-allocation		
			k-Float ACN	re-allocation of neighbors of k-th order		
			Iterative k-Flow ACN	re-allocation of neighbors of any order		
IoT-related	Licensed Spectrum	eMTC-related			IoT	[212]
		NB-IoT-related	stand-alone operation			
			in-band operation			
			guard-band operation			
	Unlicensed Spectrum	Bluetooth-related	AFHSS scheme			
			Collaborative Spectrum Allocation Scheme			
		Zigbee-related	DSSS			
		LoRa-WAN-related				
		SigFox-related				
	Both Licensed and Unlicensed	Ambient Backscatter Communication				
New LTE/NR Frequency Sharing	Semi-static				All*	[268]
	Dynamic					

* All comprises wireless, optical, IoT, edge and cloud domains.

Table A.8: Classification of Centralized Sharing Models.

Type				Domain	Works
Infrastructure Sharing	RAN Sharing	C-RAN		Wireless	[287, 311]
		Game-theory based BTS Sharing			[206]
		EC-RAN			[296]
		Resource Broker-based Schemes			[325, 6, 317, 277]
	IoT-related sharing	MBSS	Digi Device Cloud	IoT	[284]
			Sentille		
			Libelium		
			IoTSense		
			Sensor Rush		
		Caching and DL Resources Sharing			[285]
Spectrum Sharing	TV White Spaces			All*	[215]
	Centralized Network Control and Coordination Framework				[291, 345]
Sharing Among Network Slices	ICIC				[317]
	Network Slice Brokering				
	Spectrum Sharing				
	VNF Placement Consideration				
	Authentication				

* All comprises wireless, optical, IoT, edge and cloud domains.

to maintain due to the significant network heterogeneity. Lastly, the hierarchical control plane, having low-level and high-level controllers, combines benefits such as the simplicity of the implementation and the reduced latency, from both centralized and distributed architectures [7]. In the following, we discuss both distributed and hierarchical models of control and management entities within decentralized sharing models.

A.5.1 Decentralized Sharing Models

Infrastructure sharing Infrastructure sharing is a well investigated topic in wireless networks. Based on the deployed resource control and management architecture, RAN sharing can be performed either as a distributed (Distributed Radio Access Network (D-RAN)) or centralized RAN (C-RAN). In order to enable and support multi-tenancy in FCNs, Kostopoulos et al. [287] note that D-RAN requires sharing of the legacy RAN infrastructure, as well as the whole or parts of the core network. Much earlier, Frisanco et al. [2] presented details of different sharing models according to the part of the infrastructure that is about to be shared in 3GPP. The Multi-Operator Radio Access Network (MORAN) realizes sharing of active RAN infrastructure (i.e., BTSs and BSCs in 2G, as well as eNodeBs and RNCs in 3G and LTE), allowing network operators to maintain their independent control over their traffic and its QoS. With the arrival of the third generation of communication networks (i.e., 3G), another solution for sharing active RAN infrastructure - MOCN was proposed by Frisanco et al. [2]. It represents an extension to MORAN, adding the possibility of frequency pooling. In particular, each network operator possesses its own core network (e.g., EPC in LTE), which is connected to a shared Evolved Universal Terrestrial RAN (eUTRAN) via the S1 interface [249]. Given additional cost savings of frequency pooling, MOCN shows its superiority over MORAN. Stemming from MORAN and MOCN, and exploiting the synergy between SDN and NFV, the FlexRAN platform [310] follows a decentralized principle, having two main

components: FlexRAN control plane and FlexRAN Agent API. While each eNodeB has an Agent API installed, the control plane is organized in a hierarchical manner, distributing control decisions from Master Controller to each Agent. As already discussed in Section A.4.1, the hardware elements in the future core networks are envisioned to be functions that can be virtualized [7, 315, 314, 311] and thus shared. Furthermore, GWCN enables sharing of the Mobility Management Entity (MME) entity, allowing the core network to be shared as well.

Although originally presented much earlier, an alternative approach to the "conventional" sharing of RAN cells is studied by Beckman and Smith [302]. They argue that benefits can be obtained by distributing the RF power from the operators' BTSs via common shared Distributed Antenna System (DAS), usually made up of analog broadband radio repeaters and optical fibers. Thus, they clearly point at its potential to reduce CapEx and OpEx, which is not exploited enough due to the absence of network sharing.

An important decentralized model for sharing resources toward 5G networks is presented in [7]. Zhang [7] developed auction-based and contract-based algorithms for virtualization that can run in SDN controllers. In the model the InPs act as sellers, MVNOs act as buyers and SDN controllers are used to manage the virtualization process as well as signaling, forwarding, and pricing. The so-called regional controller - which executes the long-term optimization, and local controllers which provide short-term optimization in network are elaborated in great detail in [7]. Another decentralized SDN NFV-based approach to resource sharing, this time in dynamic wireless backhaul networks, is presented by Lun and Grace [340]. In order to establish balance between scalability and system performance, Lun and Grace [340] present a hierarchical architecture with two tiers of SDN controllers. In this way the communication burden is offloaded from one central to multiple local logically distributed controllers. In their multi-tenant scenario, Lun and Grace tested a resource sharing algorithm, demonstrating that their proposed architecture results in up to 40% of energy savings compared to a centralized scenario while maintaining satisfactory levels of QoS [340].

In the optical domain, Ali [224] devised a two-layered management architecture for sharing resources in terms of: i) sharing back-up path resources, ii) sharing regenerators among back-up paths. The whole sharing procedure is governed by the intermediary switching nodes. Thus, for every shared object in the network, a sharing table is employed, containing an identification of the object as well as a list of numbers for unique optical fibers. Although two different types of tables are utilized for channels and OEOs, the constraints in Ali's approach are directly related to its scalability, because of the sharing tables can become excessively large. Another example of infrastructure sharing for protection purposes is that introduced by Ruffini et al. , in [346] for converged access/metro networks. Considering a nation-wide deployment of Long-Reach PON [347], the authors devised a mechanism, based on a geometrical network coverage technique, to share backup optical transceivers across the entire country. The mechanism is based on the pre-planned disconnection of selected transceivers, which trigger a fast protection mechanism that enables load balancing, by sharing a failure across devices located in different parts of the network. Their fast protection mechanism was also experimentally demonstrated in [348].

Turning to the IoT ecosystem and its sustainability within FCNs, we briefly point out several significant attempts to share resources in this environment, in a distributed manner. In order to cope with the challenge of energy consumption in constrained IoT devices, we have

already referred to [283], in which Kouvelas et al. have proposed to share energy between IoT devices, that can be receivers and providers but not at the same time. Tackling the management structure of their solution, several control/management nodes are distributed among the entire IoT ecosystem. While numerous approaches to share resources in IoT environments are strictly theoretical, Pagani and Mikhaylov [303] presented one of the rare attempts to practically approach sharing in WSNs. Their sharing model includes dynamic discovery, negotiation, and sharing of tasks and resources between neighboring heterogeneous IoT nodes, allowing each of them to discover, request, and reserve other nodes' resources in a distributed fashion.

Yildirim and Tatar [284] also present two decentralized approaches to resource sharing: NoBV and NeBV. Their comparison of NoBV and NeBV with a centralized middleware-based model (which will be further discussed later) brings up some interesting differences between decentralized and centralized approaches, that can be considered of general validity. In NoBV virtualization is performed at each node, which is desirable for time-critical applications due to the short response time. In NeBV the authors also adapt the network virtualization protocol to the type of network considered. However, compared to centralized models, they both suffer from excessive energy consumption at the decentralized nodes, which are typically energy constrained in IoT environments.

Spectrum sharing Beside the extensive overview of infrastructure sharing models, we pay special attention to those use cases which tackle spectrum sharing, from various perspectives. Hence, authors in [214] gather all the advanced sharing models, such as D2D, IBFD, NOMA, LTE-U and CR on top of them, and present their features and potential for deployment within 5G networks. Furthermore, Khan et al. [291] extend previously published lists of advanced sharing models with License Assisted Access (LAA), Licensed Shared Access (LSA), LTE-Wi-Fi Aggregation (LWA), and Multefire. However, the authors accentuate that these models are coarse-grained and thus not suitable for achieving significant improvements in spectrum utilization efficiency.

Furthermore, two decentralized sharing trends can be recognized in the optical domain: statistical spectrum sharing and dynamic cooperative spectrum sharing. As a representative of the first one, Wang et al. [260] introduce dynamic modification of channel capacity between base and peak rates, flexibly mapping the client traffic onto an arbitrary number of universal line cards in order to compose the optical superchannel which supports the required data rate. On the other hand, dynamic spectrum sharing is extensively studied in [263], presenting a spectrum expansion/contraction policy. The concept of such sharing is considered dynamic because the policy takes into account the spectrum allocation of the neighbouring connections which compete among each other for the same spectrum resources. In fact, when a request for spectrum resources arrives: i) the relevant spectrum expansion procedure is invoked, ii) in case there are no available spectrum slots in the largest expansion region, the spectrum re-allocation procedure is triggered, 3. if spectrum re-allocation is not allowed, the request is refused [263]. Based on the feasibility of each of these three steps, Palkopoulou et al. [263] define several different dynamic spectrum sharing models, such as: Constant Spectrum Allocation (CSA), Dynamic Alternate Direction (DAD), Avoid Close Neighbors (ACN), Shift ACN, and Float ACN. They evaluate the performance of the proposed models defining case studies, conducted using Deutsche Telekom reference network

[263]. Lastly, Stiakogiannakis et al. [263] extend previous study with the following models: k-Float Blocking Neighbors and Iterative Float Blocking Neighbors.

Finally, an important distinction between spectrum sharing in IoT and conventional networks is presented in [212]. The spectrum sharing models used in conventional communication networks are mainly designed for DL long-packet communication, which is contrary to the mostly UL short-packet traffic of IoT. Thus, conventional sharing models cannot be reused for IoT applications. Another fundamental difference lays in different capabilities of the devices used in conventional and IoT networks. The conventional mobile devices are much more resourceful than IoT devices, designed with strong signal-processing capabilities and rechargeable batteries. With these differences in mind, and in order not to overload IoT devices, Zhang et al. [212] emphasize the importance of adopting simple techniques when designing spectrum sharing models for IoT. They propose a set of sharing models suitable for licensed and unlicensed bands separately, together with models that can be utilized in both licensing regimes. Interesting to notice is that there is a certain overlap in these models, since CR, NOMA, D2D, and LTE-U can be used either for conventional or IoT networks.

A.5.2 Centralized Sharing Models

Typically, sharing models where a mediator is interposed between the sharing actors and the pool of shareable resources, are characterized by higher latency and signal overhead. In this section, we present various sources which study centralized sharing models and tackle the aforementioned disadvantages, with some of them striving to prevent service disruptions potentially caused by existence of the intermediary node.

Infrastructure sharing With regard to the infrastructure sharing, we refer to several important publications and their main contributions. Using the game theory, Bousia et al. [206] propose sharing of BTSs under unrealistic assumption of a non-competitive multi-tenant scenario, in which no network operator acts selfishly and/or greedily. Nevertheless, in other to save energy and decrease expenses, redundant BTSs are being switched off upon decisions made at an arbitrarily-defined central point. However, this sharing scheme also assumes that roaming costs are low, otherwise the operators would be less likely to switch off underutilized BTSs and revert to roaming. In the context of network slicing, the Cell-Slice architecture is proposed by Kokku et al. [349], providing a gateway-level solution for slice-specific resource virtualization that impacts the individual BTS scheduling decision.

For the purpose of RAN sharing, Kostopoulos et al. [287] propose an approach to use C-RAN to improve sharing of eNodeBs. C-RANs are based on the disaggregation of eNodeBs, physically separating the RRH devices consisted of RF elements and the BBU that carries out all baseband digital processing functions. In particular, RRH devices are usually employed to extend the coverage of BTSs and eNodeBs, which are located in challenging environments (e.g., tunnels, rural areas, etc.). The two are typically connected using a Common Public Radio Interface (CPRI) protocol operating over optical fibre. When virtualized, the BBU can run as software over General Purpose Processors (GPPs) servers, located in a central office or in the cloud (BBU pool). Such virtualization enables sharing of computational

resources, as the BBUs, hosted in virtual machines or containers (e.g., Linux, Docker), can be dynamically migrated over different physical hardware.

From a perspective of RRH distribution among MNOs, Narmanlioglu and Zeydan [350] propose hierarchical SDN-based C-RAN architecture, having a RAN controller to control eNodeB functions, and a virtualization controller which performs core network sharing. In particular, they propose an RRH assignment based on load balancing algorithm for sharing RRH resources among MNOs, executed on the top of C-RAN controller. Such algorithm assigns RRHs to a particular MNO based on the number of connected UEs, unlike the traditional RRHs distribution which homogeneously distributes available RRHs. The results presented in [350] show that such load-balancing aware approach outperforms traditional RRH distribution, enabling more efficient RRHs usage. However, as resource sharing might cause insufficient isolation between operators, Niu et al. [351] present a multi-timescale dynamic resource sharing mechanism with a given level of isolation in order to decrease interference between RRHs. The output of their algorithm proves it to be robust under user mobility, while achieving the service isolation and efficient resource sharing among service providers.

An advanced version of C-RAN is the EC-RAN, designed for the stringent QoS requirements for augmented reality applications in 5G-enabled vehicular networks [296]. The EC-RAN combines C-RAN and cloud computing, and consists of numerous cloudlets which are geographically distributed to support local vehicular services.

A similar, although generalized, resource sharing architecture is presented by Kunst [325]. The resource broker is defined as a centralized entity, which is constituted of three interconnected levels: i) update level, ii) resources level, and iii) decision level. The update level is in charge of parameter collection across the whole multi-tenant network, consisting of multiple network operators which share resources. Furthermore, the resource level provides information about all available resources, while the decision level takes care of resource leasing requests and takes into account adequate pricing mechanisms and resource availability. Another example of such resource broker-related approaches, is provided by Samdanis et al. [6] with the design of an on-demand capacity broker, which facilitates on-the-fly resource allocation, thus allowing InP to allocate given portions of network capacity to an MVNO, Over The Top (OTT) operator, or any vertical market player. The layered architecture for sharing RAN and edge resources presented by Shantharama et al. [311], so-called LayBack, disseminates all resources into three layers (i.e., device layer, radio node layer, and gateway layer) which are jointly managed by an SDN orchestrator that implements SDN-based management framework in a centralized fashion, thereby coordinating the cooperation between different wireless operators and technologies. Since the SDN orchestrator decouples fronthaul from backhaul, fronthaul resources can be shared among different sharing parties. We close the elaboration of centralized infrastructure sharing in wireless domain by pointing at inter-slice sharing frameworks, which are in line with those previously elaborated.

Within the sphere of resource broker solutions, in the optical domain, a resource sharing model is presented by Raza et al. [277]. Their centralized RAN architecture with hierarchical SDN control plane is characterized by the presence of a global orchestrator, that performs sharing and optimization of resources. They show how adopting the concept of dynamic resource sharing to a limited pool of optical resources that can be shared among BTSs, results in considerable savings in overall cost of network ownership. This result was obtained

by both simulating and emulating shared network environments.

As we have already mentioned, in Section A.5.1, that the IoT-related centralized solution provided in [284] proves superior to the decentralized NoBV and NeBV approaches, we now further explore this aspect. Yildirim and Tatar [284] present their sharing model which is based on MBSS, but with significant improvements in comparison with traditional MBSS-based models. In order to prevent increase in delay and volume of signaling-related traffic, they rely on bringing the client evaluation entities closer to the shared resources. To that goal, they place client evaluation to the MBSS as the closest location. This approach requires to implement and execute the client algorithms under the same software framework. The detailed description of such software framework is provided in [284].

Considering that WSNs will become an indispensable part of 5G networks, due to the omnipresence of smart cities and their massive exploitation in FCNs, Vo et al. [285] attempt to address issues related to the limited resources of WSNs by provisioning adequate assistance from other network devices with stronger processing potential. Thus, they have designed a joint caching and DL resource sharing optimization framework, which exploits the caching storage of all existing Macro BTSs (MBSs) and Femto BTSs (FBSs), as well as the DL resources of control units in 5G networks. We associate this sharing framework to the group of centralized sharing schemes, since MBS performs collection and optimization procedures of all system parameters and then deploys the framework to cache the multimedia content in the proper FBS and to share the DL resources between the control units.

Spectrum sharing Within the topic of spectrum sharing, we shortly present three approaches which differ in philosophy as well as in the period of time when they were studied. One of the first radio bands to be considered for sharing was the TV White Space (TVWS), the broadcast channels which are unused in a certain geographic area and during a certain period of time. One approach to determine unused TV channels relies on spectrum sensing, but it was quickly recognized that in order to reliably detect incumbent TV stations the sensing threshold must be set below the noise floor. Alternatively, the FCC requires geolocation capable secondary spectrum users which then need to communicate with the TVWS databases to determine available channels. Due to excessive interference protection margins however, the potential for spectrum reuse is not fully exploited [215].

To achieve efficient and elastic spectrum utilization among multiple operators in LTE networks, Shrivastava et al. [345] designed a centralized SDN Controller, which acts as a resource brokering entity with global resource knowledge. Their approach assumes that heterogeneous LTE environments consists of Frequency Division Duplex (FDD) macro-cells, accompanied by multi-tenant Time Divison Duplex (TDD) pico-cells, allowing spectrum sharing across both. Having a TDD frame reconfiguration algorithm that dynamically adjusts UL/DL ratio for pico-cells, the trade-off between resource utilization and bandwidth is treatable and customizable. The preliminary results presented in [345] show how their SDN-based architecture significantly reduces DL delay of both FDD macro-cells, and TDD pico-cells.

Recently, Khan et al. [291] recognized the potential for fine-grained spectrum sharing aimed at achieving very stringent requirements for spectrum utilization efficiency in 5G networks. In particular, this can be realized if micro-transactions of spectrum are carried out among

Table A.9: Sharing Challenges.

Group		Type			Domain	Works
Technical	General	Abstraction of resources			All*	[7, 306, 301, 16, 265] [256, 271, 315, 310, 322]
		Isolation among operators	Fully			
			Limited			
		Isolation granularity				[7, 315, 310, 322]
		Efficient resource utilization				[306, 301]
		Customization among operators				
		QoS requirements				[252, 339]
		Required signal strength				
		Required CapEx and Opex				
		Compatibility				[249]
		Interoperability	Additional losses caused by connecting equipment of different operators		Wireless	[265, 2, 249, 238]
			Risk of incompatibility between manufacturers of eNodeBs and RNCs			
		Security			All*	[265, 264, 269, 314, 322, 324]
		Privacy				[264, 269, 314, 322, 324]
		Heterogeneity	Hardware			[264, 265, 16, 33]
			Operating systems			
			Programming languages	Programming style		
		Electromagnetic compatibility			Wireless	[249, 265]
		Access and safety during installation of shared equipment			All*	
		Deployment schedule for operators				
		Maintenance and monitoring				
		Mobility of sharing entities			IoT	[264]
		Longer response time in centralized sharing solutions			All*	[284]
	Spectrum-related	Technical complexity caused by significant difference between operating frequency domains	Linearity of power amplifiers		All*	[2, 249]
			Different antenna design requirements		Wireless	
		Coverage			All	[268]
		Wideband spectrum availability vs coverage			Iot	
		Spectrum utilization vs UL/DL coverage balance				
		TDD DL/UL switching period				
		Seamless coverage vs deployment invest				
Non-technical		Government regulations			All*	[7]
		Operators' negotiations				
		Trust among operators				[7, 249]
		Competition	Enabled competition among operators			[238, 249]
			Concurrency between sharing entities			[264]

* All comprises wireless, optical, IoT, edge and cloud domains.

network tenants, while a centralized spectrum management application controls the overall sharing from a higher perspective [291].

A.6 Challenges in Sharing Resources

Despite the undeniable benefits of sharing of network resources and recent developments in its implementation, there are indeed plenty of significant challenges remaining to be addressed.

Our presented literature review consists of numerous publications, which study sharing of network resources in various manners, and from the most diverse perspectives. According to the challenges that we have recognized by studying the literature, Table A.9 reassembles, to the best of our knowledge, all relevant sources describing various challenges related to sharing of heterogeneous resources. As Table A.9 clearly depicts, we group all sharing challenges into two non-overlapping categories, based on their technical vs. non-technical nature. In the technical category, we pay attention to the general challenges which impact both wireless and optical domains, supported by the overview of the challenges related to the IoT, edge, and cloud. Furthermore, we elaborate on the challenges which are specific for spectrum as a shareable resource and then consider non-technical challenges, such as government regulations, operators' negotiations, trust, and competition. Interestingly, we found that most of the challenges are common to wireless, optical, edge, and cloud domains.

Heterogeneity Nowadays, communication networks are characterized by highly heterogeneous types of devices, hardware equipment and platforms, radio access and backhaul technologies, configuration interfaces, actors, etc., all coexisting and cooperating in order to meet the most stringent service requirements. Silva et al. [264] and Kliem et al. [265], for instance, perceive heterogeneity as one of the main challenges that has to be overcome in the IoT world. Similarly, Taleb et al. [33] discuss heterogeneity in the context of dynamic service provisioning over distributed edge networks as a part of 5G networks.

For example, trying to exploit mechanisms derived from cloud computing [265], which usually includes pools of homogeneous resources, in the context of IoT is problematic, due to the need for each user to be able to handle any type of device [265]. Although virtualization techniques should enable tolerance to heterogeneity by enabling abstraction and isolation of resources, this comes at a price, as further discussed below.

Abstraction and Isolation Virtualization is probably the main technique to enable seamless resource sharing in 5G and FCNs, as previously discussed in subsection A.4.2. The two indispensable terms and yet inseparable from virtualization, *abstraction* and *isolation* represent the key challenges in implementing sharing models in FCN scenarios. Isolation can be considered in the context of: i) isolation of resources in general, and ii) specific isolation among network slices. As defined by Liang and Yu [16], isolation should in general ensure that any change in configuration, customization, or topology should not affect other coexisting parts of the network. Similarly, *slice isolation* refers to the cases where any failure or security attack on one network slice does not cause consequences on regular operation of other network slices [271].

Li et al. [306] consider isolation from the perspective of network operators, with a specific focus on the impact that one operator has on other operators, while sharing the same resources. Regarding the customization among operators in XG-PONs, Li et al. [306] emphasize the importance of operators being able to implement their desired scheduling algorithms, independently of the other VNOs.

Moreover, Zhang [7] and Liang and Yu [16] point at the differences between abstraction and isolation of physical resources between wireless and wired networks in general. These two virtualization procedures are particularly challenging in the wireless domain since they cannot

be easily implemented due to the fact that the wireless channel is inherently broadcast and with stochastic fluctuations [7, 16]. Liang and Yu [16] further elaborate on the undesirable properties of wireless networks, such as time-various channels, attenuation, mobility, broadcast, etc., with a special focus on cellular systems. They convey within their survey that any change in one network cell can cause significant interference to the neighboring cells, making isolation even more difficult and complicated [16]. Their comprehensive elaboration of virtualization as a sharing technique, together with the challenges and details of implementation can be found in [16].

The way in which physical resources are abstracted (and the granularity of their isolation) directly impact the efficiency of resource utilization. According to DeLeenheer et al. [256], a complete isolation is wasteful in terms of resource utilization. Their results confirm that intelligent isolation can lead to substantial savings and improved resource utilization, due to the fact that total isolation usually leads to overprovisioning of resources. The latter occurs simply because of resources being separately allocated to different network slices. In addition, having a smaller number of isolated virtual networks affects control plane scalability, because the number of control plane messages increases with the number of nodes in the network [256]. Their approach to reduce message exchange rate can be generalized and used as a template to address similar problems in other networking domains.

Isolation granularity *Isolation granularity* refers to how precisely are the resources committed to a given slice defined, impacting the level of aggregation of services or customer data into the same slice. Accordingly, Zhang [7] defined four levels of isolation, which are, from coarser to finer: i) *spectrum-level slicing*, ii) *infrastructure-level slicing*, iii) *network-level slicing*, and iv) *flow-level slicing*.

The first, coarsest level, aggregates all services delivered through a certain frequency band into the same slice. The associated methods thus simply target spectrum-level isolation. The infrastructure-level slicing instead, within a given spectrum band, assigns infrastructure resources (e.g., antennas, BTSs, backhaul, etc.) to a slice, across a shared infrastructure owned by an InP. Slicing on the network level is based on the virtualization of the whole, i.e., end-to-end network, including RAN, core network, and computing nodes within a close geographical area. Thus, all network resources are exposed in the form of packages tailored for different sharing actors and their users' demands. Within the last level, InP forms a slice of virtual resources (e.g., traffic flow), and provide it to MNOs and MVNOs. Such slice contains resources gathered with a fine granularity, and can be formed based on specific service-level requirements, such as data rate, bandwidth, latency, etc.

Spectrum-related challenges Given the characteristics and requirements for 5G-specific IoT technologies, such as eMBB and uRLLC, Wan et al. [268] discuss various challenges related to extending available spectrum to mmWave bands, and sharing UL frequency of LTE FDD frequency band as a supplemental UL carrier in the TDD band above 3GHz. In this approach, challenges are represented by trade-offs between requirements that have to be reconciled and adjusted to the service requirements. For instance, the trade-off between wideband spectrum availability and coverage in 5G is a concern, since bands below and above 3GHz reflect reciprocal relationship between coverage and data rates. The greater coverage

in bands below 3GHz leads to its limited availability and lower bandwidth, which significantly constrains achieving high data rates. On the other hand, despite high data rates, bands above 3GHz suffer from significant propagation loss which reduces coverage.

The other spectrum-related challenge is also critical and refers to the balance between efficient spectrum utilization and TDD DL/UL coverage. Simply increasing the number of slots for UL up from the one slot which is currently adopted in 5G NR, does not solve the issue. Since the UL traffic is not in balance with the DL, additional slots will increase the UL coverage but significantly decrease the spectrum utilization efficiency. Wan et al. [268] also discuss the problem that dynamically switching between TDD DL and UL would create, because of the additional delay this introduces. Finally, their consideration spans the trade-off between seamless coverage and investment into deployment [268], as the signal propagation is exposed to significant losses above 3GHz, and thus additional sites and cells will be required. It is questionable whether the operators are ready to invest more in order to enhance coverage at such high frequencies.

Interoperability The attempt to address challenges related to interoperability among different sharing actors is provided by Meddour et al. [249]. The authors address constraints related to active and passive infrastructure sharing. Active sharing entails various challenges such as those with design and configuration of antennas, since linearity of power amplifiers significantly varies across different frequency bands. According to our classification in Table A.9, such variation belongs to the spectrum-related challenges, since it is caused by operation in various frequency bands. Furthermore, the additional signal losses when interconnecting equipment of different operators, different demands on antenna design, and potential risk of incompatibility between manufacturers of eNodeBs and RNCs might lead to additional technical complexity in sharing of network resources [249]. On the other hand, Electromagnetic Compatibility (EMC) of sites, access, and safety during the installation of equipment on the shared sites, as well as deployment schedules for different operators, maintenance and monitoring of sites, are representatives of challenges in passive infrastructure sharing [249].

Security and privacy Since resource sharing includes different sharing parties (e.g., MVNOs, MNOs, InPs, SPs, different verticals, etc.) it is inevitable to preserve security and privacy requirements that are specific to each of these parties. Security is typically achieved through authentication, access control, and integrity assurance [264]. As an example, Silva et al. [264] present a detailed vision of security in sharing of resources among constrained and unconstrained devices, which provides two security levels. The more restrictive security level requires the encryption of the whole communication channel between constrained devices and endpoints, giving unconstrained devices the role of gateways with no permission to access the data. In case of less restrictive level of security, the unconstrained devices are provided with certain level of permission to access some critical resources from the sharing platform. Another example is the cloud computing-based IoT ecosystem proposed by Kliem et al. in [265] which uses Public Key Infrastructure (PKI) for this purpose.

Specifically in 5G networks, network sharing and network slicing might incur various security and privacy issues due to the transparency in operation of any sharing paradigm. Therefore,

Barakabitze et al. [269] point at the necessity for development of new 5G security and privacy protocols, which maintain the security and privacy mechanisms among slices, while enabling higher security and privacy granularity, i.e., per slice that serves various sharing actors. Furthermore, Afolabi et al. [314] and Caballero et al. [322] point at security vulnerabilities that might arise from exposing resources in multiple network slices for sharing among different tenants. As Afolabi et al. [314] provide a specific focus on VNF sharing, the higher the level of sharing VNFs between tenants, the more likely are the security vulnerabilities [314]. Thus, Afolabi et al. bring into focus the service description project [352] that is developed for network slicing, proposing the use of additional quantitative or qualitative parameters to distinguish the levels of security required by individual slices. Although each slice must have independent security mechanisms, and even a more granular approach by enabling security mechanisms on the VNF level, Caballero et al. emphasize the importance of a multi-level security framework that defines policies for different slices in multiple administrative domains, in order to prevent unauthorized access to slice resources. The lack of such framework remains a barrier towards adopting multi-tenancy approach in network slicing and sharing [322]. A specific cross-domain focus in security is brought by Taleb et al. [324], pointing at opportunities of extending border security protocols among different administrative domains that are orchestrated by multi-domain service management.

Furthermore, although their approach refers to IoT, Silva et al. [264] address the general problem related to privacy: privacy must be ensured regardless of the specifics of resource sharing. Facilitating sharing on a proprietary device (IoT or not) brings risks in maintaining privacy. Therefore, in whichever way the resources are shared, the privacy of the users who are involved in sharing must not be compromised. Despite its huge importance as it directly impacts sharing actors, privacy has not been addressed widely and requires more work.

In the last ten years, blockchain technology gained significant attention, since it avoids a single point of failure, and the security bottleneck by storing a copy of database file at the premises of all sharing entities [273]. In particular, given the opportunity to generate and use multiple keys, blockchain allows users to retain and enjoy more privacy by chaining data with hashes and pairs of keys. As Rawat [273] pointed out, trust in blockchain is established due to the group consensus where transactions are authorized by all users in the network. Accordingly, there is a significant potential in blockchain to be leveraged by resource sharing, as it enables sharing copies of transaction records to all parties, i.e., sharing actors maintaining their own instance of the blockchain database.

Non-technical challenges One of the primary goals of national regulators is to ensure competition among network operators (and/or other actors) [238], since it usually motivates operators to strive toward assuring better service quality as well as a pool of plentiful services and applications for the end users. However, such competition is tightly coupled with trust among operators, in particular when dealing with traffic monitoring and management of shared resources [238].

From the perspective of other market players (i.e., in addition to the operators), Silva et al. [264] study the competition which emerges from cooperation and sharing between constrained and unconstrained devices. Their work indicates that the instances of devices from the same pool (i.e., unconstrained/constrained) should assist their neighbors

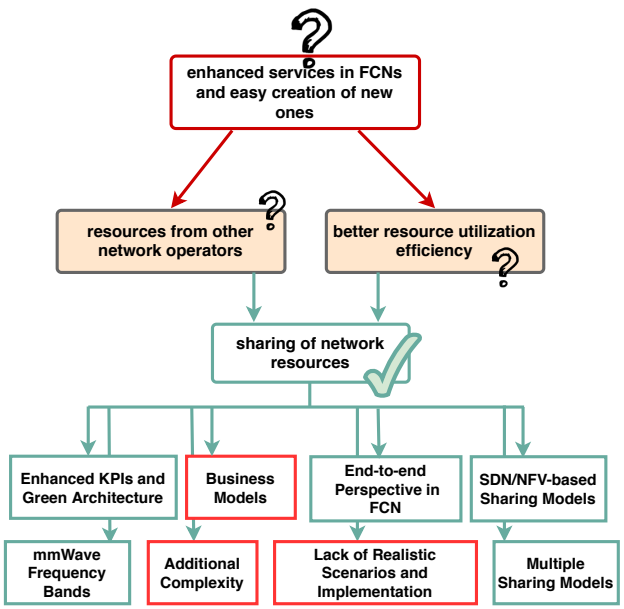


Figure A.9: The idea of sharing network resources in a nutshell; Summarized benefits and open questions.

and share with them the resources provided by devices from the other pool (i.e., constrained/unconstrained). However, the authors also press to limit this kind of assistance in order to avoid exhausting of available energy of both types of devices. All of the above open issues provide a reason to urgently tackle the need for a new business model tailored to FCNs, which defines fair strategies of sharing and assures benefits for all sharing entities.

A.7 Discussion and Open Questions

Based on the overview shown in Fig. A.9, in this section we summarize the survey and discuss some of the questions that remain to be addressed. In Fig. A.9, the green boxes reflect the beneficial nature of resources sharing. Moreover, the red boxes highlight the topics that we think require more considerable study. Due to the dynamic and challenging environments of 5G networks, it is important to enable joint operation across both existing and new services, despite their substantial differences. Thus, FCNs need to enhance existing services, while being capable of properly utilizing the full 5G potential (i.e., enhanced spectral and network efficiency, smart security, self-driving cars, enhanced QoS and QoE). As presented in Fig. A.9, there are two recognized paths that network designers and operators can take to achieve such goals and be able to cope with utmost stringent service requirements in FCNs. During the network planning and design phases, none of the operators can fully envision the amount of resources needed for proper service operation. Given the fluctuating nature of wireless traffic, the previous problem becomes even more severe, leaving the operators with excess or shortage of resources. If not properly shared, a large portion of network resources that belong to a certain network operator would remain unutilized. From the overall elaboration

provided in this survey, here we discuss and point at the the topics that either can be used as incentives for sharing or that need to be further addressed.

Enhanced KPIs and green architecture The idea of sharing heterogeneous network resources basically means releasing those resources and temporarily leasing them to other entities/actors, e.g., while not in use. There are several challenges, as presented in Section A.6, which still undermine the feasibility of resource sharing in real implementation scenarios. Nonetheless, sharing brings huge benefits in terms of enhanced KPIs and *green* network operations. The latter directly refers to the energy efficient FCNs, resulting in lower energy consumption which is particularly important in IoT scenarios, with devices with limited battery life. Some of the attempts to decrease energy consumption and thus increase the energy efficiency are presented in the survey, addressing energy sharing among devices in an IoT ecosystem as well as turning off the BTSs when traffic is low.

According to the various references studied in this survey, sharing of resources can lead to substantial savings if a resource orchestrator manages the sharing process between the slices. On the other hand, enhancing overall resource utilization by reducing the resource wastage potentially increases the possibility to accommodate even more operators in the same network. Therefore, if more operators coexist, it ultimately leads to increase in competition, which can further result in enriched and enhanced set of services for the end users. Focusing on the requirements of operators as well as users, this is beneficial for both, since increased demand for new enriched services also brings higher revenues for operators. However, achieving the optimal level of sharing resources is necessary in order to make a desirable trade-off between QoS/QoE and reduction of costs by decreasing the amount of infrastructure resources, and thus has to be studied more carefully. Furthermore, the government and environmental regulation bodies should enforce resource sharing, as they improve environmental and aesthetic conditions, as a result of lower number of locations occupied for installing network equipment, MBSs, FBSs, etc. This is particularly important for regulating 5G networks, whose high densification will introduce a significantly larger number of small cells and BTSs.

Better interrelation between business and technical models Our ability to further elaborate on the coexistence between the business, geographic, and technical models, in Section A.3.1, was limited by the lack in the literature of references that tackle them jointly. This might be justified by the fact that traditional business models tend to give operators the roles of owners of all network resources and do not include sharing as an option. Regardless of the opportunities and benefits, the real implementation of any architecture for resource sharing might not be even possible if an adequate business model is not generated in accordance with the regulative framework. Such regulatory issues were recognized long ago but still trigger the need for suitable business models, that do not limit the feasibility of the technical models. Although the formal business models are out of the scope of our work, we want to at least emphasize their importance for the proper implementation of technical models. Based on that, operators should rethink their deep-rooted business models in order to evolve from owning to sharing of resources, and to align it with the actual regulation framework.

End-to-end perspective in FCN Given the fact that 5G networks will be service-oriented, on-demand, and highly heterogeneous, there is a strong requirement to view, design, and optimize the network from an end-to-end perspective. In order to keep the 5G promises and to best serve stringent service requirements, it is essential to have an overview of all resources from wireless, optical, IoT, edge, and cloud domains, thereby spanning RAN, core network, and backhaul.

The idea to observe trends and sharing processes from such broad perspective is triggered throughout recognizing the same or similar trends in all domains, at the same or different period of time. Although sharing of the core network used to be ambiguous due to the control functions being designed around operator's ownership, some advances are recently brought together by adopting SDN and virtualization. That explains the shortage in attempts to study and approach resource sharing in core networks, particularly around service differentiation and confidentiality, which needs to be kept within one operator's boundaries. In accordance with the SDN paradigm, while the data plane ultimately releases parts of the core network for sharing, at the same time the control plane remains unshared. Sharing the data plane of the core FCN enables service differentiation, while maintaining the operator's confidentiality. To the best of our knowledge, such broad perspective adopted in our survey differs from those in existing literature, which focus on one network domain and only specific types of resources. Thus, our survey aims at facilitating future research across diverse domains, enabling their convergence, where suitable.

SDN and virtualization as enablers for future sharing The recently proposed sharing frameworks based on virtualization and SDN are quite broad and thus widely exploited for sharing resources in different domains. In particular, the main function of such sharing frameworks is to establish multiple virtual network instances, by splitting network elements into logically independent units running over the same physical substrate. These logically independent units can be further shared between different actors. Furthermore, the control architecture of the SDN/NFV framework directly impacts the sharing process and its outcomes, and it was in a greater detail discussed in Section A.5. Generally a hierarchical approach is favorable in optimizing the trade-off between complexity of the control entity and QoS/QoE levels. The control entity should consist of low-level and high-level controllers or resource brokers, combining benefits from both centralized and decentralized architectures. Another trade-off that deserves further attention in SDN/NFV enabled sharing is the balance between resource isolation and utilization efficiency in multi-tenant scenarios. Proper resource isolation is challenging, as it was discussed in the previous section, but rather important for the operators to retain control of the resources, among which are those released for sharing. Such control is inevitable for operators in order to maintain adequate levels of security and privacy.

On the other hand, a complete isolation can imply a negative effect on resource utilization efficiency since it might significantly affect the sharing ability.

The potential which lays in mmWave bands 5G networks are about to open new spectrum bands such as mmWave at frequencies between 30 and 300 GHz, which can provide novel opportunities for spectrum sharing. The disadvantage of severe attenuation could be

exploited to reuse frequencies within short distance [353], enabling cell densification. At the same time, higher densification will lead to higher sharing, in order to lower cost of network ownership. According to the FCC, the larger bandwidth available at such frequencies could potentially be competitive with fiber optics in the access network, or used jointly with fibre to provide additional resilience. Nevertheless, the deployment of services on such high frequencies has to be studied in depth due to upcoming challenges, such as 5G band selection and the unbalance between wideband spectrum availability and high data rates, the unbalance between UL and DL coverage, new investments in denser cell deployments, etc. More detailed information on the topic can be found in the FCC's proceeding [354].

Additional complexity Regardless of the way in which it is implemented, adding sharing functionality to the control and management plane of the network infrastructure increases its overall complexity. Thus, it is essential to address the trade-off between complexity and KPIs' improvements enabled by sharing. In particular, additional complexity will result in deployment of additional equipment, which can increase costs and thus offset the sharing benefits. Another source of complexity, relative to SDN is the increase in delay and signaling traffic caused by centralized architectures. As it was elaborated in the Chapter, some researchers proposed solutions consisting in moving client evaluation entities (i.e., command/queries, data aggregation and data fusion algorithms, etc.) closer to the shared resources [284] or else balancing the tasks between the SDN controller and the BTSs. In general, within the scope of FCNs, the scalability issues related to increase in complexity for network sharing requires further study.

Lack of realistic scenarios Within the literature we examined, we found several sharing models and architectures. However, there is a notable lack of realistic scenarios in their implementation, since the vast majority of the sharing models have either only theoretical foundation or their testing and validation results are obtained in a simulated environment. Apparently, the lack of adequate tools motivates researchers to extend the existing simulators or to implement new ones. This might lead to a large number of model-specific tools and software platforms which cannot be used in different environments. Taking into consideration the number of publications that we studied during preparing this survey, we realized that there is a significant lack in realistic approaches. But, there are only few attempts to mimic the real environment for the implementation of sharing resources, and we mention them here, as they might be useful to understand what can be already tested in a more realistic manner. Despite the theoretical base of their sharing approach, Kouvelas et al. [283] examined measurements from 280 households as a part of a large IoT environment. The idea for sharing energy inside the microgrid network was initiated from such real scenario. Furthermore, the cost of designing a full virtual optical mesh network topology was illustrated on a sample Italian network in order to evaluate the sharing mechanism in [224]. Indeed, the only attempt to implement sharing resources known to the authors is provided by Vilalta et al. [11]. In that case, the virtual optical network resource broker for EONs is incorporated into resource management algorithms which are evaluated in a corresponding testbed environment. The resource broker was in charge of managing virtual elastic optical resources and deploying virtual optical networks on the shared physical infrastructure. Their experimentation in the testbed confirmed feasibility of the proposed algorithms. Another realistic approach which primarily includes experimenting on testbeds, although here resource sharing is intended in

the more general sense of testbed federation, is recently presented by Both et al. [355]. Their solution encompasses multiple geographically distributed testbeds, used to orchestrate resources and to automatically scale services across multiple domains (wireless, optical, and cloud) [355].

Multiple sharing models Throughout this Chapter, we have emphasized how FCNs are envisioned to be strongly heterogeneous in technologies, devices, equipment, operators, etc. Thus, it is essential to find a way to harmonize sharing processes end-to-end and fulfil demands for services in wireless and optical domains, altogether with IoT, edge, and cloud paradigms. All of the studied approaches presented in the literature focus on either only one of the domains, or even more specifically they focus on the particular technology or service.

The aim to achieve harmonized sharing of resources with a single sharing model deployed in the network is too ambitious and highly challenging, and thus it is reasonable to consider the deployment of multiple sharing models operating in a joint manner. In particular, sharing models have to be tailored to the specific wireless and optical technologies, and especially to the IoT, edge, and cloud environments. Since all of the aforementioned areas are characterized by different requirements, single sharing models can be merged into multiple sharing model and deployed under the same software framework. An important and promising approach to support diverse experimental scenarios across multiple domains and testbeds was introduced in the previous paragraph. Namely, Both et al. [355] introduced inter-domain and inter-technology Control Framework to bridge the gap between optical, wireless, and cloud domains, enabling orchestration of diverse network resources.

A.8 Summary

As 5G networks are consisted of distributed and heterogeneous resources, with vertical services and EdgeApps that impose stringent QoS requirements, network operators are incentivized to share their network resources in order to answer excessive service demands. Thus, in this Chapter, we have presented our survey on sharing network resources, thereby discussing current and past trends in resource sharing as well as the existing tendencies to share resources in both wireless and optical domains, with specific insights into IoT, edge, and cloud paradigms. We have presented a comprehensive taxonomy with the overview of shareable resources, existing sharing techniques as well as challenges, which have to be studied more prominently in order to make resource sharing more applicable to 5G and beyond networks. The taxonomy presented in this Chapter helps to understand all the processes included in the resource sharing, and as such, it enables opportunities to design comprehensive sharing models for FCNs. Such sharing models are expected to empower the research communities to design and build more efficient next generation communication networks.

After having learned about the remaining challenges for resource sharing in 5G and beyond ecosystems in Section A.7, we focused further on exploiting the potential of NFV and SDN as cornerstone technologies for managing the resources in a more flexible and dynamic way, which was the focus of this thesis (Chapters 3-6).