An NFV System to Support Adaptable multi-UAV Service Deployments

by

Borja Nogales Dorado

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

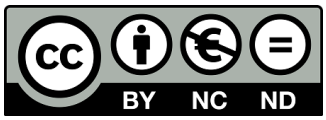
Telematic Engineering

Universidad Carlos III de Madrid

Advisor:

Dr. Iván Vidal

May 2022



"Talk is cheap. Show me the code."
— Linus Torvalds

"Don't count the days, make the days count."
— Muhammad Ali

"We cannot solve our problems with the same thinking we used when we created them."
— Albert Einstein

Acknowledgements

At first, I would like to express (although there are no words that can accurately reflect the following emotion) my sincere and complete gratitude to my supervisor, and tenured professor, **Iván Vidal**. His dedication and guidance, from the very beginning with the final bachelor's project back in 2016 until today, have been fundamental to accomplish the numerous achievements on which the elaboration of this thesis has been based. The countless and priceless hours in his office discussing possible developments and research directions, in addition to the time shared on project journeys, have greatly contributed to the sublime fondness I feel for this wonderful (and at the same time, challenging) stage.

Special gratitude goes to **Francisco Valera**, better known within our small group as the *Jedi Master*. His exceptional support and wise advice, always constructive, have served to enhance the quality of each and every one of the works included in this thesis. In the same bag is my partner-in-crime, **Víctor Sánchez**, who has the merit of making me feel welcome in the department from the very first minute of my arrival. This, together with the hours and hours shared over the last 5 years (on and off the university campus), turned him into a great friend and a fellow soldier. Others responsible for the special affection I have for this stage, and who deserve a special place in this acknowledgements section, are my office colleagues: **Jorge, José, Lewis** and **Antonio**. To them goes the great merit of making the long days, tackling arduous research tasks, enjoyable, and many of them (if not all of them) amusing. In this regard, and because of the many much-needed coffees shared to refresh minds, a well-deserved acknowledgement goes to the department's colleagues Kiril, Milan, Sergio, Ginés, Pelayo, Patricia, Don Oscar, Donato, Guimarães, Winnie, Marco, Stefano, Cristina, Gonzalo, Miguel, and Jesús.

My deepest gratitude goes to my family, especially my parents **Antonio** and **Mª Teresa**, whose unconditional love, trust and support throughout the years of my existence have served as a catalyst for the human being I am today. Special mention also goes to my two brothers, **Raul** and **Isaac**, who have fulfilled their role as big brothers flawlessly, and to my niece and nephew **Nayara** and **Enzo**. They are the ones responsible for enlivening the child that lives in me. I can not forget Juanjo and Daniel, who as chosen family, are fundamental characters in every episode.

Last but not least, I would like to dedicate this work to **Irene**, my beloved and future wife. She has been, and continues to be, the unceasing light in my darkest moments. Thank you infinitely for being my company, and for holding my hand. I have no doubt that together we will be able to face every obstacle we encounter along the way. I love you.

Published and Submitted Content

In compliance with the principles referring to plagiarism in Law 14/2011 and in the Code of Good Practices of the Universidad Carlos III de Madrid (UC3M) Doctoral School, I hereby report a bibliography of articles or other contributions I have (co)authored that are included as part of the thesis and that have been published or submitted for publication.

Articles published in conferences:

- **B. Nogales**, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A NFV system to support configurable and automated multi-UAV service deployments," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications - DroNet'18*, Munich, Germany, 10–15 June: MobiSys 2018, Association for Computing Machinery, 2018, pp. 39–44. DOI: 10.1145/3213526.3213534.

  - The content of this work is explicitly referenced in Chapter 4;
  - The role of the author of this thesis relates to the preliminary design and prototype of a system that encompasses Unmanned Aerial Vehicles (UAVs) to support the provision of moderately complex applications and services by means of the Network Functions Virtualization (NFV) technology;
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

- **B. Nogales**, I. Vidal, V. Sanchez-Aguero, F. Valera, and L. F. Gonzalez, "An NFV system to support service provisioning on UAV networks," in *JITEL 2021: Libro de actas: XV Jornadas de Ingeniería Telemática*, A Coruña, Spain, 27–29 October, 2021, p. 228.

  - This work is partly included and its content is reported in Chapter 8;
  - The role of the author of this thesis relates to the analysis on the future standardization challenges of considering a wide range of heterogeneous devices with computational resources (*e.g.*, user equipment terminals, or UAVs) that might exist in a particular deployment area, and be

opportunistically integrated within an NFV infrastructure. Thus, supporting the operation of cost-effective and reliable services beyond the network access segments of telecommunications operators;

– The material from this source included in this thesis is not singled out with typographic means and references.

Articles published in indexed journals and magazines:

- **B. Nogales**, V. Sanchez-Aguero, I. Vidal, and F. Valera, "Adaptable and Automated Small UAV Deployments via Virtualization," *Sensors*, vol. 18, no. 12, p. 4116, 2018. DOI: 10.3390/s18124116.

  – This work is wholly included and its content is reported in Chapter 4;

  – The role of the author of this thesis relates to the analysis of the European Telecommunications Standards Institute (ETSI) NFV framework in the context of the UAVs, and to present the initial design of the system proposed in this thesis to support flexible, automated and cost-effective deployment of network services over small-sized UAVs. In addition, it includes the implementation of a functional prototype, as well as the execution of experimentation activities defined to validate the system through a realistic use case, which involves the deployment of an IP telephony telco service;

  – This work was published in the *MDPI Sensors* journal, which was ranked within the Q1 of Journal Citation Reports (JCR) with an impact factor of 3.031;

  – The material from this source included in this thesis is not singled out with typographic means and references.

- **B. Nogales**, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso, and A. Azcorra, "Design and Deployment of an Open Management and Orchestration Platform for Multi-site NFV Experimentation," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 20–27, 2019. DOI: 10.1109/MCOM.2018.1800084.

  – This work is wholly included and its content is reported in Chapter 3;

  – The role of the author of this thesis relates to the realization of the design, deployment and validation of the open source NFV Management & Orchestration (MANO) platform built at 5TONIC in a time frame in which open-source NFV solutions were emerging, and the deployment of an NFV ecosystem with the ability to integrate vertical infrastructures was a subject of great interest;

  – This work was published in the *IEEE Communications Magazine* journal, which was ranked within the Q1 of JCR with an impact factor of 11.052;

  – The material from this source included in this thesis is not singled out with typographic means and references.

- **B. Nogales**, I. Vidal, V. Sanchez-Aguero, F. Valera, L. F. Gonzalez, and A. Azcorra, "Automated deployment of an Internet protocol telephony service on unmanned aerial vehicles using network functions virtualization," *JoVE (Journal of Visualized Experiments)*, no. 153, e60425, 2019. DOI: 10.3791/60425.

  - This work is partly included and its content is reported in Chapter 4;
  - The role of the author of this thesis relates to the definition of a protocol to configure a network functions virtualization environment using UAVs as computational entities providing the underlying structure to execute virtualized network functions. This protocol also includes the procedures to configure an elaborated IP telephony service, showcasing the the multi-site capability of the system presented in the previous work;
  - This work was published in the *JoVE (Journal of Visualized Experiments)* journal, which was ranked within the Q3 of JCR with an impact factor of 1.163;
  - The material from this source included in this thesis is not singled out with typographic means and references.

- I. Vidal, **B. Nogales**, F. Valera, L. F. Gonzalez, V. Sanchez-Aguero, E. Jacob, and C. Cervelló-Pastor, "A multi-site NFV testbed for experimentation with SUAV-based 5G vertical services," *IEEE access*, vol. 8, pp. 111 522–111 535, 2020. DOI: 10.1109/ACCESS.2020.3001985.

  - This work is wholly included and its content is reported in Chapter 5;
  - The role of the author of this thesis relates to the utilization of the system presented in the previous works to create a multi-site testbed at national scale, supporting the realization of 5G vertical oriented services based on UAVs. It also includes the definition and development of a smart-farming vertical service, as well as the validation activities driven by the practical experimentation with a use case related to the precision agriculture;
  - This work was published in the *IEEE Access* journal, which was ranked within the Q2 JCR with an impact factor of 3.367;
  - The material from this source included in this thesis is not singled out with typographic means and references.

- **B. Nogales**, L. F. González, I. Vidal, F. Valera, J. García-Reinoso, D. R. López, J. Rodríguez, N. González, I. Berberana, and A. Azcorra, "Integration of 5G Experimentation Infrastructures into a Multi-Site NFV Ecosystem," *JoVE (Journal of Visualized Experiments)*, vol. e61946, 2021. DOI: 10.3791/61946.

  - This work is partly included and its content is reported in Chapter 3;
  - The role of the author of this thesis relates to the definition and experimentation of a detailed protocol, based on previous experiences with the management of multi-site NFV infrastructures, to support the flexible incorporation of 5G experimentation infrastructures into a multi-site NFV

ecosystem. For this purpose, the protocol includes the complete procedure to utilize an overlay network architecture based on Virtual Private Networks (VPNs);

  - This work was published in the *JoVE (Journal of Visualized Experiments)* journal, which was ranked within the Q3 of JCR with an impact factor of 1.355;
  - The material from this source included in this thesis is not singled out with typographic means and references.

- **B. Nogales**, M. Silva, I. Vidal, M. Luís, F. Valera, S. Sargento, and A. Azcorra, "Using Aerial and Vehicular NFV Infrastructures to Agilely Create Vertical Services," *Sensors*, vol. 21, no. 4, p. 1342, 2021. DOI: 10.3390/s21041342.
  - This work is wholly included and its content is reported in Chapter 6;
  - The role of the author of this thesis relates to explore the integration of the small-sized UAVs system (proposed in the previous works) with other similar research efforts in the field of vehicular networks. As a result, the role of the author also included the implementation of an NFV framework capable of integrating aerial and vehicular NFV infrastructures, enabling the cost-effective and flexible deployment of vertical services. In addition, it included the detailed description and realization of a public-safety vertical use case, emphasizing the practicality and potential benefits of the proposed framework, that was deployed by instantiating the constituent components in NFV sites of different countries (Spain and Portugal);
  - This work was published in the *MDPI Sensors* journal, which was ranked within the Q1 of JCR with an impact factor of 3.576;
  - The material from this source included in this thesis is not singled out with typographic means and references.

- I. Vidal, **B. Nogales**, D. Lopez, J. Rodríguez, F. Valera, and A. Azcorra, "A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services," *Electronics*, vol. 10, no. 15, p. 1868, 2021. DOI: 10.3390/electronics10151868.
  - This work is partly included and its content is reported in Chapter 7;
  - The role of the author of this thesis relates to the realization of the prototype and experimentation activities of a novel platform to support secure link-layer connectivity for virtual functions within a multi-site NFV ecosystem, as well as its validation through the execution of an IP television (IPTV) service based on the multicast technique. It is worth noting that this work builds on previous work, with the idea of addressing the limitations that exist in the common approaches used to support multi-site communications in NFV environments;
  - This work was published in the *MDPI Electronics* journal, which was ranked within the Q3 of JCR with an impact factor of 2.397;
  - The material from this source included in this thesis is not singled out with typographic means and references.

Submitted articles:

- **B. Nogales**, I. Vidal, V. Sanchez-Aguero, F. Valera, and D. R. Lopez, "Software-driven overlay networks for inter-site communications in NFV cross-domains", In: *IEEE Communications Magazine*, [Submitted] May, 2022.
  - This work is wholly included and its content is reported in Chapter 7;
  - The role of the author of this thesis relates to the baseline analysis, design, implementation and validation of a framework that evolves the secure link-layer connectivity platform presented in the previous work. For this, the role of the author also includes the integration of the Software Defined Networking (SDN) technology within the connectivity platform to enable the creation of software-driven overlay networks, and thus, exploiting the possible options available to link heterogeneous infrastructures within a multi-site NFV ecosystem;
  - The material from this source included in this thesis is not singled out with typographic means and references.

Contribution to Standards Development Organizations:

- **B. Nogales**, I. Vidal, V. Sanchez-Aguero, F. Valera, L. F. Gonzalez, and A. Azcorra, *OSM PoC: Automated Deployment of an IP Telephony Service on UAVs using OSM*, [Online] Available: https://osm.etsi.org/wikipub/index.php/OSM_PoC_10_Automated_Deployment_of_an_IP_Telephony_Service_on_UAVs_using_OSM, (accessed on May. 17, 2022), 2020.
  - Participant in the development of the Proof of Concept (PoC) submitted to ETSI within the Open Source MANO (OSM) technology, which aims at demonstrating the practical feasibility of automating the deployment of telecommunication services over resource-constrained devices, particularly UAVs. In addition this PoC was awarded as the best PoC with OSM during the eighth development release cycle of OSM.

- J. Ordonez-Lucena, D. R. Lopez, M. Xie, P. Grønsund, A. J. Gonzalez, C. Guerrero, **B. Nogales**, I. Vidal, A. Gallego, S. Denazis, D. Giannopoulos, P. Papaioannou, Y. Chatzis, C. Tranoris, and K. Trantzas, *ZSM PoC: Automated network slice scaling in multi-site environments*, Presentation at ETSI BrightTalk Channel, [Online] Available: https://www.etsi.org/events/1905-webinar-zsm-poc-2-showcase-automated-network-slice-scaling-in-multi-site-environments, (accessed on May. 17, 2022), 2021.
  - Contributor to the development of the PoC submitted to ETSI within the Zero touch network & Service Management (ZSM) technology, which aims to demonstrate the capability to automatically scale out a deployed network instance across multiple administrative domains.

- J. Ordonez-Lucena, D. R. Lopez, D. Camps, A. Fernández-Fernández, G. Bernini, P. Giardina, T. Cogalan, A. Mourad, **B. Nogales**, I. Vidal, and F. Valera, *ZSM PoC: On-demand Non-Public Networks (NPNs) for industry 4.0: zero-touch provisioning practices in public-private network environments*, [Online] Available: https://zsmwiki.etsi.org/index.php?title=PoC_5_On-demand_Non-Public_Networks_(NPNs)_for_industry_4.0:_zero-touch_provisioning_practices_in_public-private_network_environments., (accessed on May. 17, 2022), 2022.

  – Contributor to the development of the PoC submitted to ETSI within the ZSM technology, which intends to showcase the ability to provision a zero-touch (*i.e.,* with no human intervention), and on-demand 5G private network for accommodating industry 4.0 services.

Other Research Merits

This chapter first provides a list of additional publications I have (co)authored, which are related to this thesis but not included therein. Next, it provides an overview of the various European and Spanish funded projects where I was involved in during the lifetime of this thesis, as well as my role and participation. Finally, it includes other significant achievements that were accomplished under the scope of this thesis.

Related publications:

- I. Vidal, P. Bellavista, V. Sanchez-Aguero, J. Garcia-Reinoso, F. Valera, **B. Nogales**, and A. Azcorra, "Enabling Multi-Mission Interoperable UAS Using Data-Centric Communications," *Sensors*, vol. 18, no. 10, p. 3421, 2018.

- V. Sanchez-Aguero, **B. Nogales**, F. Valera, and I. Vidal, "Investigating the deployability of VoIP services over wireless interconnected micro aerial vehicles," *Internet Technology Letters*, vol. 1, no. 5, e40, 2018.

- A. P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. Silva, I. Vidal, **B. Nogales**, *et al.*, "5GinFIRE: An end-to-end open5G vertical network function ecosystem," *Ad Hoc Networks*, vol. 93, p. 101 895, 2019.

- V. Sanchez-Aguero, F. Valera, **B. Nogales**, L. F. Gonzalez, and I. Vidal, "VENUE: Virtualized Environment for Multi-UAV Network Emulation," *IEEE Access*, vol. 7, pp. 154 659–154 671, 2019.

- L. F. Gonzalez, I. Vidal, F. Valera, V. Sanchez-Aguero, **B. Nogales**, and D. R. Lopez, "NFV orchestration on intermittently available SUAV platforms: challenges and hurdles," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2019, pp. 301–306.

- L. F. Gonzalez, I. Vidal, F. Valera, **B. Nogales**, V. Sanchez-Aguero, and D. R. Lopez, "Transport-Layer Limitations for NFV Orchestration in Resource-Constrained Aerial Networks," *Sensors*, vol. 19, no. 23, p. 5220, 2019.

- C. Tipantuña, X. Hesselbach, V. Sánchez-Aguero, F. Valera, I. Vidal, and **B. Nogales**, "An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

- W. Nakimuli, J. Garcia-Reinoso, **B. Nogales**, I. Vidal, D. Gomes, and D. Lopez, "Reducing Service Creation Time Leveraging on Network Function Virtualization," *IEEE Access*, vol. 8, pp. 155 679–155 696, 2020.

- V. Sanchez-Aguero, I. Vidal, F. Valera, **B. Nogales**, L. L. Mendes, W. Damascena Dias, and A. Carvalho Ferreira, "Deploying an NFV-based experimentation scenario for 5G solutions in underserved areas," *Sensors*, vol. 21, no. 5, p. 1897, 2021.

- V. Sanchez-Aguero, F. Valera, I. Vidal, **B. Nogales**, J. Cabezas, and C. Vidal, "A virtualization approach to validate services and subsystems of a MALE UAV," in *IEEE INFOCOM Workshops: The 9th International Workshop on Computer and Networking Experimental Research using Testbeds*, IEEE, 2022.

Participation and role in National and European research projects:

European H2020 5GinFIRE Project

5GinFIRE (Evolving FIRE into a 5G-Oriented Experimental Playground for Vertical industries) was a research project under the EU programme Horizon 2020, which started on January 2017. The primary goal of the project was to establish an NFV-enabled experimental testbed for 5G technologies, capable of instantiating and supporting vertical industries based on industry-leading and open source technologies. The overall experimental testbed in 5GinFIRE was distributed across different locations, including Spain, Portugal, United Kingdom and Greece, and other facilities across Europe that were integrated throughout the project. The project also included an additional facility located in Brazil. Being framed under the EU FIRE initiative, the project embraced the execution of more than 25 experiments, which were conducted by several third parties (experimenters and 5G experimental facilities owners/operators), and selected through the planned 5GINFIRE Open Calls.

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since January 2017 to the end of the project (December 2019);
- Contribution to the identification and analysis of the main open source projects and technologies, relevant in the field of NFV, in order to build the reference experimentation platform;
- Perform the deployment and maintenance of the NFV site hosted in the 5TONIC laboratory, located in Madrid;
- Participate in the definition of the mechanisms for supporting the interconnection of vertical infrastructures, as well as in the incorporation of all these infrastructures;
- Design and implement new orchestration functionalities for the OSM upstream project, as well as different functions and services, that were contributed to the community as open source;

– Develop enhancements and solutions to address inconveniences confronted during the operations of orchestration platform;
– Analyze the integration of a particular NFV infrastructure based on UAVs, with limited computing resources, within the orchestration platform;
– Support and assistance during the realization of the experiments accepted during the 5GINFIRE Open Calls;
– The content included in Chapter 3 and Chapter 4 is related to the scope of this project.

European H2020 5GRANGE Project

5GRANGE (Remote area Access Network for 5th GEneration) was an European research project under the EU programme Horizon 2020, which involved Brazilian universities and research entities, with the aim of designing, manufacturing, and validating the mechanisms to enable the 5G network to provide an economically effective solution for Internet access in remote areas. Thus, the developed remote area access network for 5G can be used to provide reliable and cost-effective Internet access in low populated areas, as well as enable new agribusiness services and support high mobility coverage for highways. In addition, the project also explored different cost-effective approaches to extend the network infrastructure offered by the 5GRANGE access network, *e.g.* to support network communications beyond the boundaries of the radio cells. In particular, 5GRANGE considers UAV devices, which may be deployed over delimited geographic areas, as well as other ground units that may be opportunistically available within that areas (*e.g.,* harvesters and tractors).

The role and the activities of the author of this thesis in the project are the following:

– Actively participating in the project since its kick-off in November 2017 to the end of the project (October 2020);
– Research, prototyping and experimental validation of an NFV-based UAVs infrastructure capable of extending the access network defined within the context of the project to support voice and data communications services;
– Demonstration of an IP telephony telco service leveraging the NFV-based UAVs infrastructure;
– Participate in the implementation of the PoC submitted to ETSI within the OSM technology, entitled "Automated Deployment of an IP Telephony Service on UAVs using OSM";
– The content included in Chapter 5 is related to the scope of this project.

European H2020 5G–VINNI Project

5G-VINNI (5G Verticals INNovation Infrastructure) project started in July 2018, after its selection by 5G Infrastructure Public Private Partnership (5G-PPP) commission to promote implementing and testing advanced 5G infrastructures in Europe. The project was committed to leverage the latest 5G technologies, including results from previous 5G-PPP phases, employing advanced network virtualization, slicing, radio and core technologies. In addition, an automated validation effort was defined to validate 5G under the amalgamation of different technologies and network loads. Specifically, 5G-VINNI was

hosted by four main sites located in Norway, United Kingdom, Spain and Greece. Then, two experimental sites located in Germany and Portugal were integrated.

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since January 2021 to the end of the project (December 2021);
- Advise on the design and implementation of the inter-site communications that enabled the operations within the end-to-end 5G facility (composed of several inter-working sites), which were similar to the mechanisms defined within the 5GinFIRE project;
- Participate in the implementation of the PoC submitted to ETSI within the ZSM technology, entitled "Automated network slice scaling in multi-site environments".

Spanish 5GCity Project

5GCity (Adaptive Management of 5G Services to Support Critical Events in Cities) was a Spanish research project, funded by the Spanish Ministry of Economy and Competitiveness, which started in January 2017 with the main objective of addressing one of the most critical challenges of 5G networks: the provision of telecommunications services in high-density urban areas, mainly driven by the potential growth to be expected in the context of smart cities. In order to demonstrate the social impact of the proposed solutions, 5GCity focused on a situation involving a relatively large number of mobile users concentrated in a small area, resulting in traffic concentrations due to congestion or accidents, disasters, or emergency situations. In this context, the project strategy intended to develop an adaptive solution to coordinate different 5G technologies and establish communication services, while ensuring robust scalability and high reliability, and providing very low latency when necessary.

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since January 2017 to the end of the project (June 2019);
- Contribute to the creation of a NFV ecosystem at national scale, which was used to validate the SUAVs-based NFV infrastructure within the scope of vertical services;
- Demonstration of a smart-farming vertical service based on precision agriculture, involving the NFV-based UAVs infrastructure;
- The content included in Chapter 4 and Chapter 5 is related to the scope of this project.

Spanish TRUE5G Project

TRUE5G (Towards zeRo toUch nEtwork and services for beyond 5G) is a Spanish research project funded by the Spanish National Research Agency in 2019, and oriented to support the applications and services provisioning of the forthcoming mobile network generations (*e.g.*, automated driving, precision agriculture, environmental awareness, smart-health, or connected-industry, to name a few). For this purpose, TRUE5G considers the creation of novel algorithms, technologies, and mathematical artifacts to promote the advanced system configuration and automation envisioned within the zero-touch paradigm.

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since its kick-off in 2019 to date;
- Participate in the elaboration of an NFV framework capable of integrating aerial and vehicular NFV infrastructures, to enable the cost-effective and flexible deployment of vertical services;
- Research and design of a novel platform capable of enabling the secure data-layer communications among VNFs that are placed on distributed NFV environments (*i.e.*, inter-site communications);
- Prototyping and demonstration of the platform with the execution of an IPTV service;
- Design and implement the evolution of the secure link-layer connectivity platform using the SDN technology, supporting software-driven overlay networks to link the infrastructures of a multi-site NFV ecosystem;
- The content included in Chapter 6 and Chapter 7 is related to the scope of this project.

European H2020 LABYRINTH Project
LABYRINTH (Unmanned traffic management 4D planning technologies for drone swarm to enhance safety and security in transport) is an European research project whose fundamental objective is focused on ensuring airspace safety through an air-traffic management system for unmanned aerial vehicles (or as these devices are most commonly known, drones). With this system, LABYRINTH aims to avoid potentially problematic situations such as those that arise when drones fly over civilian environments at low altitude. To this end, the project address the challenge imposed by the lack of maturity in remote guidance and control technologies. In addition, to validate the relevance of the envisioned management system, the project defines four different use cases: *(i)* road-transport pilot; *(ii)* waterborne transport pilot; *(iii)* air transport pilot; and *(iv)* emergency pilot

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since its kick-off in June 2020 to date;
- Participation in the design of architectural communications framework to support the operations of the drones within the context of the system envisioned in the project;
- Contribution to the research and analysis on the current virtualization technologies that may provide UAVs with the ability to efficiently provide different applications and services;
- Analysis of potential solutions to support secure communications between VNFs in UAVs (*e.g.*, IPsec, and/or MACsec).

European H2020 5GZORRO Project
5GZORRO (Zero-tOuch secuRity and tRust for ubiquitous cOmputing and connectivity in 5G networks), is an EU-funded project aiming to address the lack of framework enabling the pervasive share of computing, transport, radio and spectrum resources. To tackle this limitation, 5GZORRO intends

to support vertical applications and services at production-level by enabling the automation of end-to-end network slicing across multiple network stakeholders (*e.g.*, operators, and/or infrastructure providers). For this purpose, 5GZORRO identifies different technologies of interest such as: Artificial Intelligence (AI), Distributed Ledger Technology (DLT), and cloud-native technologies (*i.e.*, NFV and SDN). To validate the potential benefits of the objective outlined in 5GZORRO, the project includes the definition of three different use cases that will be deployed across the two available facilities in the project (5GBarcelona, and 5TONIC): *(i)* smart-contracts for ubiquitous computing/connectivity, *(ii)* dynamic spectrum allocation, and *(iii)* pervasive virtual Content Delivery Networks (CDNs) over 3rd-party edge resources

The role and the activities of the author of this thesis in the project are the following:

- Actively participating in the project since September 2021 to date;
- Contribution to the development of the project's platform within the 5TONIC facility;
- Tailor the 5TONIC NFV ecosystem to simultaneously accommodate the experimentation activities of different projects;
- Support to identify the needed resources to host the use cases defined within the scope of the project;
- Design and implementation of the communications model to connect the services that are spanned across the facilities of the project (*i.e.*, 5GBarcelona, and 5TONIC);
- Participate in the implementation of the PoC submitted to ETSI within the ZSM technology, entitled "On-demand Non-Public Networks (NPNs) for industry 4.0: zero-touch provisioning practices in public-private network environments".

Participation in research-oriented events:

Speaker at the OSM MR-10 Hackfest organised by ETSI, presenting the definition of an NFV framework capable of integrating aerial and vehicular NFV infrastructures to enable the cost-effective and flexible deployment of vertical services:

- **B. Nogales**, M. Silva, I. Vidal, L. Silva, F. Valera, S. Sargento, and A. Azcorra, *Realization of a public safety vertical use case based on OSM and aerial/vehicular NFV infrastructures*, [Online] Available: https://osm.etsi.org/wikipub/index.php/OSM-MR10_Hackfest, (accessed on May. 17, 2022), 2021.

Speaker at the NFV Evolution virtual event organised by ETSI, presenting the fundamental challenges posed by a vision of dynamically composing NFV infrastructures and deploying services in capacity-constrained environments, which will need to be satisfied by virtual infrastructure management platforms, or VIMs:

- **B. Nogales**, I. Vidal, V. Sanchez-Aguero, L. F. Gonzalez, F. Valera, and A. Azcorra, *An NFV system to support service provisioning on UAV platforms: a walkthrough on implementation experience and standardization challenges*, Presentation at ETSI NFV Evolution Event, [Online] Available: https://www.telecomtv.com/content/etsi-nfv-evolution-event-agenda-day1, (accessed on May. 17, 2022), 2021.

Awards and distinctions obtained:

Award for the best proof of concept (PoC) with Open Source MANO (OSM) by the European Telecommunications Standards Institute (ETSI) in recognition of the technical contribution to the practical demonstration of ETSI OSM capabilities during the EIGHT release cycle of the OSM software: https://osm.etsi.org/news-events/osm-awards

The following article was recognised as one of the ten best-valued contributions proposed by the evaluators for the “Premio ISDEFE I+D+i ANTONIO TORRES". As part of this recognition, the article was included in the Spanish Ministry of Defence publication “Los diez artículos finalistas del DESEi+d 2018”, published on 31 March 2019 (ISBN-13: 978-8490914243):

- I. Vidal, V. Sanchez-Aguero, F. Valera, **B. Nogales**, J. Cabezas, C. Vidal, A. Lopez, D. Gonzalez, J. Diez, L. Berrazueta, and M. Merino, “Milano: una visión futura para un UAS táctico,” in *Proceedings of the VI Congreso Nacional de I+D en Defensa y Seguridad*, Valladolid, Spain, 20–22 November: Ministerio de Defensa, Secretaría General Técnica, 2018, p. 167.

Abstract

A main aspect that has characterized the evolution towards the 5^{th} Generation of Mobile Networks (5G) has been the involvement of industrial sectors, or as they are generally known, verticals, in the definition of the requirements that these networks must address with respect to the service provisioning. Thus, this paradigm shift not only considers services that facilitate human communications, but also promotes the creation of a global digital ecosystem in which verticals such as the automotive sector, smart-cities, health-care, or public-safety are key adopters. In addition, this change in the service provisioning model also facilitates the appearance of innovative services (*e.g.*, virtual reality, augmented reality, healthcare, autonomous driving, etc.), which are not only expected to cause an enormous growth in the amount of traffic, but also to demand a high performance from the network.

One of the technological advances that has driven this evolution toward 5G has been the adoption of technologies for function virtualization and softwarization supporting the transition from traditional specialized hardware equipment (*e.g.*, IP routers, firewalls or load balancing devices) to versatile software components, which can be deployed in different locations. In particular, 5G embraces NFV, a technology standardized by the ETSI, that enables the automated and agile provision of telecommunication and vertical services in 5G networks, as a composition of virtualized components, commonly referred to as Virtualized Network Functions (VNFs). However, under the temporal context in which the beginning of this thesis is situated (*i.e.*, in 2017), NFV was starting to receive a great interest from the industry and research community, and there were just a few open source initiatives aiming to implement the standard. Due to this, an additional effort was needed to understand, by means of existent implementations at that time, the implications and challenges of applying the NFV standards in practical situations. In this context, the first part of this thesis is devoted to creating a platform capable of enabling complex, close to reality, experimentation scenarios across a distributed set of NFV and vertical infrastructures, which can be made available by different stakeholders at different geographic locations. Thus, consolidating important aspects of the standard, as well as identifying new necessary specifications.

In accordance to these specifications, the next part of this thesis is intended to address the lack of flexibility on the 5G networks to support reliable service operations in environments and situations where there are obvious resource constraints. For instance, in *(i)* remote areas where 5G radio access network coverage is insufficient or non-existent; *(ii)* emergency situations (*e.g.*, natural disas-

ters), where the network infrastructure may fail or provide deficient service; or *(iii)* situations where there are occasional high, unexpected or predictable, service demands such as in the case of mass events. To this purpose, this thesis explores the potential benefits of creating an NFV system based on Unmanned Aerial Vehicles (UAVs) to exploit the inherent capabilities of this aerial devices, and extend the programmable substrate of 5G networks beyond the network access segments of telecommunications operators.

Then, the thesis analyzes the potential benefits of using the UAV-based system in the deployment of services included within the context of different vertical sectors. Firstly, this part analyzes possible synergies between NFV, UAVs, and vertical services from a practical perspective, presenting the creation of a multi-site testbed at national scale to support prototyping, and experimentation activities. This testbed builds on the NFV system based on UAVs, and on the mechanisms related to the orchestration of telecommunications and vertical services within a multi-site NFV ecosystem, previously discussed in the context of this thesis. Moreover, this testbed allows to realize the practical validation related to the capability of the UAVs-based system to support vertical services. This validation is performed then with the definition of a use case involving smart-farming vertical, instantiating a precision agriculture service over the UAVs on a remote site. Subsequently, following this line, this thesis explores the interoperation of the UAV-based system with other NFV infrastructures, with the aim of supporting the deployment of telecommunications and/or vertical services in resource-constrained situations. In particular, this part considers a situation related to the public-safety vertical, where the system collaborates with an NFV infrastructure composed of a fleet of vehicles to assist a response team in the management of an emergency.

To conclude, this thesis addresses an additional key challenge to support the provision of telecommunications and vertical services in 5G: the provision of adequate mechanisms to enable the exchange of data traffic between VNFs, which may be located in different 5G domains. This is not only interesting from the general point of view of the NFV ecosystems, but this type of communications particularized for the UAVs-based system, could allow this system to collaborate with other NFV infrastructures *e.g.*, cloud/edge platforms) to support the proper operations of more elaborated services. In this context, this last part of the thesis presents a novel solution to support secure link-layer connectivity for virtualized functions in multi-site NFV ecosystems. Thus, providing an appropriate mechanism to enable the exchange of data traffic among VNFs that are located in different NFV domains. As part of this work, the thesis explores the use of a SDN framework to evolve such a solution, offering an inter-domain connectivity orchestration service, which intends to support the automated and on-demand provisioning and configuration of virtual networks between different NFV domains.

Table of Contents

List of Figures

List of Tables

List of Acronyms

2G 2^{nd} Generation of Mobile Networks
3G 3^{rd} Generation of Mobile Networks
3GPP 3rd Generation Partnership Project
4G 4^{th} Generation of Mobile Networks
5G 5^{th} Generation of Mobile Networks
5G-PPP 5G Infrastructure Public Private Partnership
5TONIC 5G Telefonica Open Network Innovation Centre

AI Artificial Intelligence
API Application Programming Interface
ATIS Alliance for Telecommunications Industry Solutions

BSS Business Support System

CCDF Complementary Cumulative Distribution Function
CDN Content Delivery Network
CI/CD Continuous Integration and Continuous Deployment
CNCF Cloud Native Computing Foundation
CNI Container Network Interface
CPE Customer Premises Equipment
CPU Central Processing Unit

DHCP Dynamic Host Configuration Protocol
DLT Distributed Ledger Technology

EDGE Enhanced Data Rates for GSM Evolution
eMBB enhanced Mobile Broadband
EPC Evolved Packet Core
ETSI European Telecommunications Standards Institute
EU European Union

FANET Flying Ad-hoc Network

GCS Ground Control Station
GPRS General Packet Radio Service
GRE Generic Routing Encapsulation
GSM Global Systems for Mobile Communications

HSDPA High Speed Downlink Packet Access
HSPA High Speed Packet Access
HSUPA High Speed Uplink Packet Access
HTTP Hypertext Transfer Protocol

IGMP Internet Group Management Protocol
IoT Internet of Things
IPsec Internet Protocol Security
ISG Industry Specification Group
ITU International Telecommunication Union

JCR Journal Citation Reports

KPI Key Performance Indicator

LTE Long Term Evolution

MANO Management & Orchestration
mMTC massive Machine Type Communications
mmWave millimeter wave

MTU Maximum Transmission Unit

N3IWF Non-3GPP Inter-Working Function
NAT Network Address Translation
NFV Network Functions Virtualization
NFVI NFV Infrastructure
NFVO NFV Orchestrator
NR New Radio
NS Network Service
NSA "Non-Stand-Alone"
NSD Network Service Descriptor
NTP Network Time Protocol
NUMA Non-Uniform Memory Access

OBU On-Board Unit
ONF Open Networking Foundation
OS Operating System
OSM Open Source MANO
OSS Operations Support System

PoC Proof of Concept
PSCF Publish–Subscribe Configuration Function

RAT Radio Access Technology
RO Resource Orchestrator
RSU Road Side Unit
RTT Round Trip Time

SA "Stand-Alone"
SBC Single-Board Computer
SDN Software Defined Networking
SDO Standard Development Organization
SLA Service Level Agreement
SMS Short Message Service
SO Service Orchestrator
SoTA State of the Art
SR-IOV Single Root I/O Virtualization
SSH Secure Shell

TCP Transmission Control Protocol

UAV Unmanned Aerial Vehicle
UC3M Universidad Carlos III de Madrid
UE User Equipment
UMTS Universal Mobile Telecommunications System
UPC Universidad Politécnica de Cataluña
UPV/EHU Universidad del País Vasco
URLLC Ultra-Reliable Low Latency Communications

V2I Vehicle-to-Infrastructure
V2V Vehicle-to-Vehicle
VANET Vehicular Ad-hoc Network
VCA VNF Configuration and Abstraction
VIM Virtualized Infrastructure Manager
VM Virtual Machine
VNF Virtualized Network Function
VNFD VNF Descriptor
VNFM VNF Manager
VoIP Voice over IP
VPN Virtual Private Network
VXLAN Virtual eXtensible Local Area Network

WCDMA Wideband Code Division Multiple Access

ZSM Zero touch network & Service Management

CHAPTER 1

Introduction

It is unquestionable that end-user preferences, as well as their utilization of the information technology systems, have undergone a major change in the last decade. Social media applications like WhatsApp or Twitter, and multimedia services such as YouTube, Netflix or Spotify have been acquiring a notable prominence within all the Internet traffic. This tendency has led to an exponential increase in such traffic over the networks worldwide. Additionally, some of the most relevant forecasts, such as [1], point to an even more significant boost in this traffic due to the massive connection of different forthcoming devices (*e.g.*, wearables, smartphones, sensors, *etc.*), and to the new emerging services brought about the new generation of communications, *i.e.*, 5^{th} Generation of Mobile Networks (5G) [2]. These services (*e.g.*, virtual reality, augmented reality, healthcare, autonomous driving, etc.) are not only expected to cause an enormous growth in the amount of traffic, but also to demand a high performance from the network. The latter is the responsible for providing higher data rates, ultra-reliable and low latency communications, reduced and optimized energy consumption, and ubiquitous network access that enable the proper operation of the mentioned services. Due to this, different stakeholders (*i.e.*, telecommunication/network operators, vendors, infrastructure providers, etc.) have been exhaustively working to make possible a disruptive change in the networking paradigm.

This paradigm shift has been delineated as the transition from a model in which the provision of communications services was exclusively oriented towards addressing human communications (where human communications refers to services enabling audio/video calls, data transmission such as e-mail, and Internet access), towards a model that also considers vertical sectors, or in other words, specific industrial sectors (*e.g.*, automotive, smart-cities, healthcare, public-safety, etc.), as key adopters. Thus, 5G is intended to create a global digital ecosystem [3], involving the unprecedented require-

ments outlined by those vertical sectors [4, 5], and to increase the portfolio of feasible products and services.

The evolution towards the 5G is being mainly driven by significant technological advances, both in terms of wireless connectivity and service provisioning models. Radio access technologies have been substantially enhanced to support differentiated configurations, in particular enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC). On the other hand, the adoption of technologies for function virtualization and softwarization is playing a fundamental role [6], supporting the transition from traditional specialized hardware equipment (*e.g.*, IP routers, firewalls or load balancing devices) to versatile software components, which can be deployed in different locations. In particular, 5G embraces Network Functions Virtualization (NFV), a technology standardized by the European Telecommunications Standards Institute (ETSI)[7], and based on both concepts. More precisely, NFV enables the automated and agile provision of telecommunication and vertical services in 5G networks, as a composition of virtualized components, commonly referred to as Virtualized Network Functions (VNFs). Likewise, NFV provides a high degree of flexibility to deploy these virtualized components in distributed NFV infrastructures (*e.g.*, in edge environments close to end-users), facilitating the achievement of the performance requirements imposed on 5G networks, and making an efficient use of the available resources in computing environments.

While these technological advances have brought about an unprecedented evolutionary leap in the field of mobile communication networks, it should be noted that the service provision model have largely preserved the use of infrastructures whose compute resources are typically connected to fixed access networks (*e.g.*, based on Ethernet technologies). For instance, despite the high degree of flexibility provided by technologies such as NFV to deploy services in different locations, in practice, such deployments are mostly carried out in cloud or edge infrastructures. This has a detrimental impact on supporting reliable service operations in environments and situations where there are obvious resource constraints such as: *(i)* remote areas where 5G radio access network coverage is insufficient or non-existent; *(ii)* emergency situations (*e.g.*, natural disasters), where the network infrastructure may fail or provide deficient service; or *(iii)* situations where there are occasional high, unexpected or predictable, service demands such as in the case of mass events.

Taking into account the above considerations, one of the pillars of this thesis is focused on exploring the possibility of using NFV infrastructures that can be deployed on-demand, beyond the network access segments of telecommunications operators. In particular, Unmanned Aerial Vehicles (UAVs) are considered for this study, since these devices could extend the programmable substrate of the 5G networks by incorporating a diverse and dynamic catalogue of computing, storage and network resources to accommodate the situations outlined in the previous paragraph. It is worth noting that the realization of the aforementioned vision is challenging since these aerial devices are commonly designed and manufactured to accomplish specific missions, and not to expose the control of their compute, storage and networking resources to be easily and dynamically configured for other services. Figure 1.1 shows a simplified representation of the UAVs to compose on-demand NFV in-

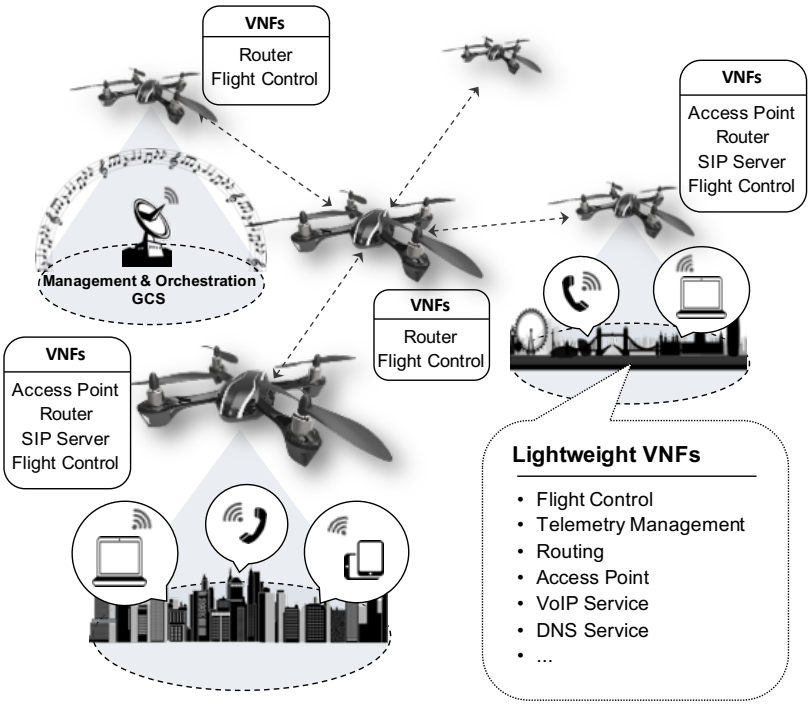


Figure 1.1. *Utilization of UAVs to compose NFV infrastructures and service deployments.*

frastructures, and deploy telecommunication and vertical services.

On the other hand, it is important to emphasize that virtualization technologies provide a large flexibility to deploy network functions in different locations, assuming that sufficient compute, storage and network resources are available to support their proper operation. This is especially relevant for the vision presented above considering the service deployments with UAVs, since different NFV infrastructures such as edge/cloud platforms, or different UAV swarms, might collaborate to realize the execution of moderately complex multi-site services.

However, this flexibility related to the placement of VNFs raises new requirements in terms of connectivity: whereas connectivity among VNFs in the same domain can be supported by creating virtual links over NFV infrastructure of the domain, providing such connectivity in a multi-domain NFV environment is problematic. This is due to the fact that NFV domains will typically be distributed in different geographic locations, and communication among them will take place through untrusted network domains of Internet service providers that, in most cases, will be external to the operations of the NFV ecosystem. Moreover, different NFV domains may be owned by different entities using different management and orchestration policies, as well as different mechanisms and tools to implement these policies (*e.g.*, different Management & Orchestration (MANO) solutions).

For these reasons, inter-domain communications in distributed, multi-site NFV ecosystems have typically relied on network layer routing mechanisms (*i.e.*, routing over the Internet) and/or the use of overlay network technologies, such as Virtual Private Networks (VPNs). This approach has demonstrated to be effective to enable secure network-level connectivity among distant NFV domains. None-

theless, it is important to note that it has significant limitations to support data communications among VNFs of the same service deployed in different domains. On the one hand, the use of network-level routing mechanisms hampers proper isolation among multi-domain NFV services. That is, in the absence of specific mechanisms to prevent this, VNFs of a service could be reachable from VNFs of other services, or even by untrusted third-parties. On the other hand, the use of this approach entails the potential need for additional (non-desirable) networking configurations in VNFs after their deployment, in order to properly participate in inter-domain network level routing operations (*e.g.*, the next IP hop of a VNF might not be another VNF in the service, but a router in the local network domain that offers connectivity to external networks). This prevents the multi-domain nature of an NFV ecosystem from being opaque to its users. Finally, it is worth noting the potential requirement to configure additional forwarding state in the underlying network infrastructures that support inter-domain connectivity. This is motivated because such networks will have to route traffic originated and/or terminated in the VNFs of the multi-domain services, which will use their own IP address space.

Given the above considerations, providing appropriate mechanisms to enable the exchange of data traffic among VNFs that are located in different NFV domains emerges as a fundamental challenge for the provision of services in 5G networks, and as an additional pillar of the thesis. In this context, this thesis aims at providing a platform for supporting secure link-layer connectivity among NFV domains, enabling the automated provision of connectivity among VNFs, and allowing to address the limitations outlined above. The initial premise of this research line is that the successful deployment of NFV services in 5G and beyond networks will necessarily require the multi-domain nature of an NFV ecosystem to be opaque to entities requesting the deployment of a service. That is, a telecommunication service or vertical must operate correctly, according to the service specification provided by the entity requesting the deployment, regardless of whether the VNFs comprising the service have been deployed in one or more NFV domains. This opacity involves the creation of multi-domain virtual networks, providing the abstraction of a link-layer local network where VNFs of the same service, deployed in different NFV domains, can automatically connect.

1.1. Research Objectives

In the light of the considerations outlined in the previous section, five general objectives were considered to cope with during the realization of this thesis, each of which has been addressed in the subsequent chapters (see Section 1.2).

OBJECTIVE O1: ANALYSIS ON THE VIRTUALIZATION TECHNOLOGIES AND THE NFV STANDARDS

The NFV paradigm has been currently consolidated as one of the key enabling technologies in the development of the 5G networks. This technology aims at alleviating the hardware dependencies in the provision of network functions and services, using virtualization techniques that allow those functionalities to be executed in commodity servers over an abstraction layer. However, under the

temporal context in which the beginning of this thesis is situated (*i.e.*, in 2017), NFV was starting to receive a great interest from the industry and research community, and there were just a few open source initiatives aiming to implement the standard. Due to this, an additional effort was needed to understand, by means of existent implementations at that time, the implications and challenges of applying the NFV standards in practical situations. Thus, this effort would consolidate important aspects of the standard, as well as identify new necessary specifications.

In this context, the main purpose of this first objective aims at analyzing the NFV ecosystem standards defined by the ETSI, as well as the main virtualization technologies and open source initiatives in the NFV arena. In addition, this objective is also intended to, on the basis of the previously mentioned analysis, design and deploy an NFV MANO platform at 5TONIC, the open research and innovation laboratory on 5G technologies founded by Telefonica and IMDEA Networks. As relevant design principles, this NFV MANO platform is based on open source technologies, and aims at incorporating external sites to complement the portfolio of software and hardware resources that can be made available for experimentation activities. Thus, providing 5TONIC with a functional production-like NFV environment, and enabling experimentation with novel NFV products and services.

OBJECTIVE O2: AN NFV SYSTEM BASED ON UAVS TO SUPPORT AUTOMATED AND ADAPTABLE SERVICE DEPLOYMENTS OVER DELIMITED AREAS

As previously commented, a fundamental aspect of this thesis is to make a first step towards the creation of an NFV infrastructure that can be deployed on-demand, beyond the network access segments of telecommunications operator. Thus, offering a flexible platform capable of enabling cost-effective and adaptable deployment of services over delimited geographic areas, where communications infrastructures are not available, nor sufficient (*e.g.*, remote areas, emergencies, etc.).

To this end, this thesis proposes the objective of exploring the potential benefits of creating an NFV system based on Unmanned Aerial Vehicles (UAVs). These aerial devices offer a promising platform for this purpose due to their inherent mobility, as well as the possibility they offer to incorporate a diverse catalogue of computing, storage and networking resources. Thus, extending the programmable substrate of 5G networks. In addition, this objective includes the design and the prototype implementation (based on open source technologies) of the NFV system based on UAVs. It is worth noting that, the realization of this system is challenging due to, among other reasons, the limited capacity of the hardware and software platforms that can typically be on-boarded on UAVs; the need to automatically manage the resources provided by that platforms and deploy virtual functions on top of them, despite being transported by aerial vehicles; or the requirement to specify appropriate placement policies for virtual functions (*e.g.*, and to indicate which virtual functions should be executed over the same UAV unit). As a final part of the objective, this involves the integration of the system within the NFV MANO platform located at 5TONIC, and the validation of the system proposed through the deployment of a moderately complex telecommunications service. Specifically, it considers an IP telephony service where different users in the vicinity of the UAVs may access, through the functionalities offered by the aerial devices, to the telephony service hosted in the facilities of a telecommunications operator.

OBJECTIVE O3: EXPLORE THE USE OF THE NFV SYSTEM BASED ON UAVS IN THE CONTEXT OF DIFFERENT VERTICAL SECTORS

After outlining the design of an NFV system based on UAVs, and demonstrating its practical feasibility with the implementation of a functional prototype and the realization of a particular use case (*i.e.*, an IP telephony service), the next objective is oriented towards exploring the potential benefits of using this system in the context of different vertical services. In particular, this objective considers two vertical applications where the characteristics of a system comprising UAVs may be of special interest: *(i)* a smart-farming vertical service; and a *(ii)* public-safety vertical service.

On the one hand, UAVs have lately gained attention within the smart-farming vertical sector due to the ability of these aerial devices to on-board different payloads such as high-resolution cameras, or low-cost sensor and electronic devices. Based on information that can be acquired by an UAV from these devices, aerial vehicles may serve multiple purposes including, among others, monitoring and remote sensing operations over crop fields. On the other hand, within the context of the public-safety vertical, UAVs are relevant since a fleet of these aerial devices could rapidly be deployed to assist in addressing an emergency through the execution of different moderately complex services. For instance, UAVs could provide voice and data communications to a team of firefighters, working on a fire extinction activity along a large forest area, where existing infrastructures may be insufficient or even unavailable. In this context, this objectives delineates and implements two different use cases to prove the feasibility of the system to offer applications and services within the scope of both verticals.

OBJECTIVE O4: A MULTI-SITE PLATFORM TO ENABLE SECURE AND RELIABLE COMMUNICATIONS AMONG VNFS DEPLOYED ACROSS DIFFERENT NFV INFRASTRUCTURES

As discussed above, a fundamental challenge to support the provision of telecommunication services and verticals in 5G is to have appropriate mechanisms to enable the exchange of data traffic between VNFs, that might be located in different NFV domains. This is particularly interesting from the perspective introduced in this thesis regarding the provision of services through the UAV-based NFV system, as it enables additional NFV infrastructures (*e.g.*, cloud/edge or other UAV swarms) to collaborate with that system in the execution of moderately complex services.

In this context, this objective is oriented to realize the analysis, and the subsequent implementation, of a novel solution capable of safely and reliably supporting the communications among VNFs that may be distributed across several NFV infrastructures. The solution is intended to automatically be deployed and configured as a regular multi-site NFV service, providing the abstraction of a layer-2 switch that offers link-layer connectivity to VNFs deployed on remote NFV sites. Furthermore, this objective encompasses the use of existing softwarization mechanisms, such as Software Defined Networking (SDN), to incorporate flexibility and programmability to the inter-domain communications solution. Finally, the objective includes a validation section, which utilizes the multi-site NFV ecosystem at 5TONIC to verify the practical feasibility of the proposed solution through the realization of a use case based on an IP television (IPTV) service.

OBJECTIVE O5: STUDY OF FUTURE STANDARDIZATION CHALLENGES

Based on the lessons learned with the work realized in the context of the previous objectives, this one aims at exploring the implications and challenges of using opportunistically a wide range of heterogeneous devices that might exist in a particular deployment area, such as user equipment terminals, Customer Premises Equipment (CPE), or any other device available in residential environments, or smart cities. This way, those devices would contribute to extend the available and programmable resources (in terms of compute, storage, and networking) to support cost-effective and reliable services beyond the network access segments of telecommunications operators. In this context, this objective is focused on carefully analyzing the challenges to be addressed for materializing the outline vision under the perspective of the NFV MANO framework.

1.2. Thesis Overview

This section outlines how the remainder of this thesis has been structured, as well as the information contained in each of the chapters that constitute this structure:

- ***Chapter 1. Introduction:*** this chapter has described the context in which this thesis is established, presenting the motivations for it, as well as the objectives to be achieved.
- ***Chapter 2. Background & Related Work:*** provides background information and State of the Art (SoTA) review of the set of technologies used for the realization of this thesis. Specifically, it includes the main aspects of 5G, the details of the NFV technology, and an overview of service orchestration over compute-constrained devices, with a special attention to UAVs.
- ***Chapter 3. Open NFV MANO Platform for Multi-Site Experimentation:*** defines the design principles of a MANO platform capable of managing and orchestrating multi-site services comprising multiple infrastructures, distributed in different geographical locations. In addition, it includes the detailed functional implementation aspects of the platform, which were exclusively based on open source technologies. With that, this chapter tackles the Objective O1, presented in the previous section.
- ***Chapter 4. Adaptable and Automated Small UAVs Deployments via NFV:*** addresses the Objective O2. For this, it carefully describes the design of an NFV infrastructure, where a set of UAVs provide the substrate in terms of computing, storage, and network to enable the deployment on-demand of moderately complex network services. Additionally, it presents a prototype implementation of the design, and perform its validation through the instantiation of an IP telephony service
- ***Chapter 5. A Multi-Site NFV Testbed for Experimentation with SUAV-Based 5G Vertical Services:*** details, from a practical perspective governed primarily governed by experimentation,

the definition of a multi-site testbed based on open source technologies to explore synergies among NFV and UAVs, to support the deployment of services oriented towards vertical sectors. Specifically, this chapter defines a use case based on the smart-farming vertical to validate the capabilities of the multi-site testbed, covering one of the verticals considered in Objective O3.

- ***Chapter 6. Using Aerial and Vehicular NFV Infrastructures to Agilely Create Vertical Services :*** following the line drawn in the previous chapter to address the Objective O3, this chapter analyzes the possible synergies with related systems. In particular, it explores the use of two particular mobile infrastructures for service provisioning: an UAVs-based NFV infrastructure that can be deployed on-demand over a specific area; and an automotive-based NFV infrastructure that may be opportunistically present in a deployment area. As a result, this chapter describes an international collaboration between the Universidad Carlos III de Madrid (UC3M) (Spain) and the Instituto de Telecomunicações of Aveiro (Portugal). With demonstrative value, this collaboration includes the definition of an elaborated use case based on the public-safety vertical, which emphasizes the main benefits of integrating these trendy mobile NFV infrastructures.

- ***Chapter 7. A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services :*** focused on addressing the Objective O4, this chapter involves the development of a novel platform to support secure and reliable link-layer communications among VNFs running in different NFV infrastructures (separated in terms of their geographical location). Thus, enabling a novel mechanism for providing inter-site communications within the NFV ecosystem through the use of overlay networks techniques. In addition, this chapter introduces the use of an SDN framework to provide the inter-site communications solution with the ability of dynamically and programmatically orchestrate the overlay networks enabling these inter-site communications.

- ***Chapter 8. Conclusions & Future Work :*** presents the conclusions derived from the lessons learned with the realization of this thesis, and the future research lines to be explored. In this context, these future lines present the study on the future standardization challenges targeted at Objective O5.

- ***Appendix A :*** provides further information about the technical specifications of the NFV MANO platform elaborated in Chapter 3, as an extension to the actual document.

CHAPTER 2

Background & Related Work

This chapter presents an overview of the main technologies that have sustained the elaboration of the content included throughout the following chapters of this this thesis. In particular, the first section of this chapter introduces the main aspects of the 5^{th} Generation of Mobile Networks (5G), identifying the new elements that this new generation brings with it, emphasizing the intended requirements to be achieved with its deployment, and pointing out the key technologies that are enabling those requirements to become a reality.

Subsequently, the principal aspects of the Network Functions Virtualization (NFV) technology are elaborated. This technology is considered as one of the most relevant technologies within the functions virtualization and softwarization category, and currently have been consolidated as a main enabler of this fifth generation of communications. Due to this, and to the major role it plays in the realization of this thesis, Section 2.2 elaborates the main aspects of the NFV technology, describing in detail the main elements that constitute its architectural reference framework. In addition, this part of the section encompasses an analysis of available implementations of this technology, as well as the details on the virtualization mechanisms on which it is based.

Finally, this chapter includes the studio of a particular component involved in the development of this thesis: the Unmanned Aerial Vehicles (UAVs). Firstly, Section 2.3 sheds light on the management of the virtual resources provided by resource constrained devices. Then, it focuses on analysing the impact of these aerial devices on 5G networks, which has recently turned out to be more than just a trendy idea. To conclude, it reviews the use of UAVs to support the service orchestration over these compute-constrained aerial devices through virtualization and softwarization mechanisms.

2.1. Fifth Generation of Mobile Networks (5G)

This section is devoted to dissecting the fundamental aspects that were considered to define the 5^{th} Generation of Mobile Networks (5G). As previously commented, the main differential change between this generation and the previous four generations, and the reason behind the disruptive paradigm shift in the arena of mobile communications, is that this generation does not pursue a mere improvement in network performance to accommodate more elaborated services for end-users (*e.g.*, access over Internet to multimedia content at higher speed ratio) [8]. In essence, this generation aims for a change in the service provision model, with the underlying assumption of a large number of connected machine-type devices, which has led to the emergence of new possibilities in the provision of services. Thus, these services can now be oriented towards specific industry verticals (*e.g.*, automotive, healthcare, public-safety, etc.) [2].

To provide a better insight on the pillars on which 5G is being founded, the following lines present a succinct summary of the objectives that have been pursued over the previous generations of mobile networks, along with their key features. Afterwards, this section outlines the Key Performance Indicators (KPIs) that were targeted for 5G deployment, as well as the key enabler technologies that were identified as candidates to achieve those KPIs.

2.1.1. Radio access technologies evolution

Radio access technologies have been constantly evolving since the inception of mobile calls in approximately 1980s, when phone calls used to be completely analogue and relying on the fixed telephone network infrastructure (*i.e.*, circuit-switching communications). Soon thereafter, the introduction of the Global Systems for Mobile Communications (GSM) technology during the 1990s revolutionised mobile communications since its specification was not only intended for the establishment of digital mobile calls through the fixed network infrastructure, but also extended to support the exchange of data messages (*i.e.*, Short Message Service, or SMS) over the network. Since that, and considering the potential of using mobile communications not exclusively for the exchange of voice data, but also the provision of additional services, the 3rd Generation Partnership Project (3GPP) was founded by a collaboration of different telecommunication associations (such as the European Telecommunications Standards Institute (ETSI) [9], or the US Alliance for Telecommunications Industry Solutions (ATIS) [10]) with the main objective of establishing the bases for a global mobile communication system.

Since its inception in 1998, the 3GPP has been the main standardization body in charge of defining the high-level specifications for radio access technologies, improving their specification to enable the provision of complex communications services over mobile networks. These technologies are nowadays supplied for the provision of data communication services (*e.g.*, for Internet access) to mobile and wireless terminals by the leading worldwide telecommunication operators and are also sustained by the main manufacturers of networking facilities and systems. Table 2.1 provides a brief overview of the evolution of the main mobile communications technologies up to the introduction of 5G, to

situate the foundations on which the fifth generation of mobile networks were built [11].

GSM	Global Systems for Mobile Communications (GSM) is a communications system that brought about the 2G. Initially developed in the 1990s to operate in the 900 MHz frequency band, it was soon adapted to serve the 1800 MHz frequency band. In addition, the GSM design has been tailored over the years to properly operate in other frequency bands. The 2G communication systems were originally intended for the voice information exchange, and they were based on the application of circuit-switched technologies. Such systems were also extended to support the exchange of data messages by means of the SMS. Moreover, several carrier services were defined with GSM to enable data transmission rates of up to 9.6 Kbps over switched circuits. All the details about the GSM radio technology are defined in the 45 series of the technical specifications defined by 3GPP [12]
GPRS/ EDGE	The need to provide an enhanced service for data transmission to mobile telecommunication network terminals led to the development of the so-called 2.5G telecommunication systems. These systems already introduced packet-switched technologies into the core of the telecommunications system, as well as modifications to the radio interface for data transmission (not just voice). Thus, General Packet Radio Service (GPRS) technology emerged in the early 2000s as an evolution to GSM. Originally, the GPRS specification allowed mobile terminals to receive data at rates up to 40 Kbps, and to transmit at rates up to 14 Kbps. Subsequent improvements would increase the theoretical download capacity up to 171 Kbps. The evolution to provide enhanced data rates was Enhanced Data Rates for GSM Evolution (EDGE), which is also known as *Enhanced GPRS*. This technology allowed data rates of up to 384 Kbps, and it was referred to as 2.75G of mobile networks. Further information on GPRS and EDGE technologies can be found in the 44 [13], and the 45 [12] series of the technical specifications defined by 3GPP.
UMTS	Universal Mobile Telecommunications System (UMTS) is an umbrella denomination that encompasses the various 3G radio technologies developed by 3GPP. UMTS complements the GSM architecture, introducing new technologies for the radio interface in the access network, in which Wideband Code Division Multiple Access (WCDMA) is particularly relevant. In the sixth 3GPP specifications release (published in 2005), WCDMA enables theoretical peak data rates up to 14 Mbps in the network-to-terminal (*i.e.*, downlink) direction, and data rates up to 5.7 Mbps in the terminal-to-network (*i.e.*, uplink) direction. The definition of UMTS technology is covered in the 3GPP specification series comprised between 21 and 37 [14].
HSPA	Successively to UMTS, and with the aim of increasing the data transmission performance, High Speed Packet Access (HSPA) appeared as an improvement to WCDMA (for this reason, HSPA is commonly referred to as 3.5G). HSPA consists of two technologies: High Speed Downlink Packet Access (HSDPA) [15], which supports peak user data rates up to 13.4 Mbps at the data link level; and High Speed Uplink Packet Access (HSUPA) [16], with a peak data rate at the data link up to 5.4 Mbps.
LTE	Long Term Evolution (LTE) is the most recent evolution of UMTS defined by 3GPP (initially in its eighth specifications release, published in 2008) with the aim of preserving the competitiveness of its 3G mobile communications system, while addressing the increasing demands of users regarding data transmission rates and quality of services. To this end, LTE eliminates the circuit-switched domain present in GSM, and introduces a new packet-switched domain denoted Evolved Packet Core (EPC). On the radio part, LTE evolves and replaces the UMTS terrestrial radio access network, as well as the radio interface with the terminal. Additionally, the LTE design allows theoretical peak data rates of 300 Mbps in the downlink direction, and 75 Mbps in the uplink direction. Later on, LTE-Advanced emerged in 2011 improving the performance offered by LTE. In particular, the system was designed to enable theoretical peak data rates of 3 Gbps in reception at the mobile terminal, and 1.5 Gbps in transmission, utilizing a bandwidth of 100 MHz (nevertheless, these performance metrics, as well as the previously indicated for LTE and WCDMA are theoretical, and unattainable in practice). Currently, LTE and LTE-Advanced are considered 4G. Both the definition and detail of the LTE technology, in the same manner as UMTS, is covered in the 3GPP specification series comprised between 21 and 37 [14].

Table 2.1. *Summary of mobile communications generations evolution up to 5G.*

2.1.2. Overall 5G Key Performance Indicators

The comprehensive vision related to the service provisioning model included in the fifth generation of mobile networks considers a wide variety of use cases related to different vertical sectors. Just to name a few, some of these include: virtual/augmented reality, smart offices, online gaming, remote computing, automotive driving, smart wearables, etc. As can be expected, each of them has different characteristics and therefore, diverse requirements. With the aim of defining a set of common requirements for the numerous use cases mentioned above, the several organizations that have been driving the definition of 5G networks, led by the International Telecommunication Union (ITU), reached an agreement to group these use cases into three main lines of work [17]:

- ***enhanced Mobile Broadband (eMBB):*** related to the user experience, this line targets peak speeds of 20 Gbps (*i.e.*, 20 times higher than 4G), and increasing traffic density by 100 times per unit area. Thus allowing, for instance, ultra high definition content or augmented reality experiences;
- ***massive Machine Type Communications (mMTC):*** this group aims to increase the capacity to support a density of connected devices 10 times higher than 4G, which will enable, among other things, the massive deployment of sensors, the Internet of Things (IoT), and the growth of services dealing with large data sets (also known as *Big Data* services);
- ***Ultra-Reliable Low Latency Communications (URLLC):*** this line encompasses low latency communications of around 1 ms compared to 20-30 ms considered for 4G networks. This condition could potentially enable applications that have stringent requirements in this area, such as connected vehicles or tele-medicine services (*i.e.*, healthcare assistance remotely).

Taking into account the considerations described above, the ITU and 3GPP conducted a collaborative effort to elaborate the set of most important performance parameters of the 5G systems (or in other words, the overall 5G KPIs), assigning nominal reference values for each of these parameters [18, 19]. Figure 2.1 summarizes these general requirements for 5G. Subsequently, other Standard Development Organizations (SDOs) formulated additional requirements of high relevance with respect key 5G key capabilities. For instance, 5G Infrastructure Public Private Partnership (5G-PPP) specified average service creation time not higher than 90 minutes, and a 90% reduction of energy consumption per deployed service [20, 21].

2.1.3. Key-enabling Technologies for 5G

As previously mentioned, the evolution towards the 5G is being mainly driven by significant technological advances, both in terms of wireless connectivity and service provisioning models. Radio access technologies have been substantially enhanced to support differentiated configurations, in particular eMBB, URLLC, and mMTC. In this context, the 3GPP laid the foundations for a new Radio Access

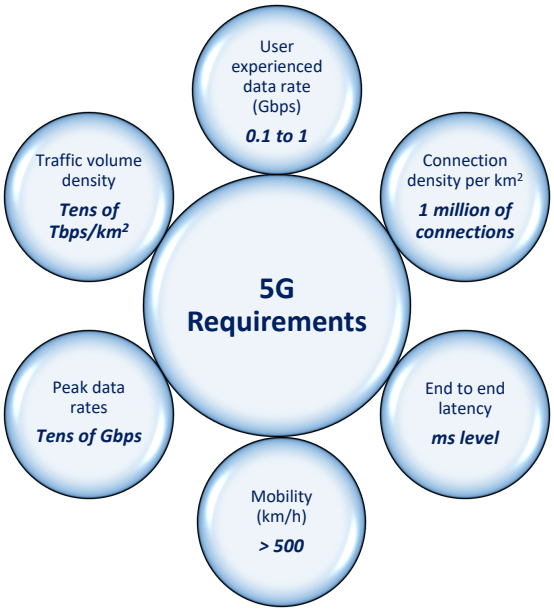


Figure 2.1. *5G requirements defined by ITU and 3GPP.*

Technology (RAT), called 5G New Radio (NR), with the aim of providing a global standard for the 5G radio interface.

The analysis of this technology started in 2015, and the first specification outlining the fundamental aspects of this technology was published in late 2017 [22]. This specification included the basics of the "Non-Stand-Alone" (NSA) NR connectivity model for 5G. This model relies on the already deployed 4G infrastructure. Specifically, the communication between the end-user terminal and the base station systems (*i.e.*, the *fronthaul* segment) is handled by mechanisms and protocols incorporated with 5G, but the network path between the base station and the core network (*i.e.*, the *backhaul* segment) requires the 4G core architecture. In this manner, this 5G NSA model has allowed operators in the last couple of years to offer an alternative option that improves the performance of 4G networks, while completing the implementation of the 5G "Stand-Alone" (SA) model. This latter is elaborated in the 3GPP Release 15 [23], which was published in 2018, and included the first complete set of standalone 5G standards.

In both of these models, the NR technology utilizes three frequency bands at European level, catering to three different environments: *(i)* the 3500 MHz band for urban environments, offering higher speed rates than those experienced with 4G (around 1~2 Gbps per user). This band will be the most widely used due to its ability to comply with the requirements exposed by end-user services; *(ii)* the 700 MHz band, devoted to rural areas due to its capacity to reach very long propagation distances. Notwithstanding the trade-off in terms of reduced data rates, this band provides an interesting substrate for communicating areas that are currently without coverage; and *(iii)* the 26GHz band, more commonly known as millimeter wave (mmWave) band, which is oriented to specific environments due to its reduced propagation and severe degradation upon physical obstacles. Nevertheless, this

band is attractive for connected industry environments, or smart cities, since it offers data rates of around 10 Gbps. This could allow the operations of diverse use cases related to vertical sectors with extremely fast data rate information exchange requirement.

In addition to the radio access technology covered with the 5G NR, the adoption of technologies for function virtualization and softwarization, in particular Software Defined Networking (SDN) and Network Functions Virtualization (NFV), has unlocked new opportunities to automate the lifecycle management of telecommunication and vertical services, efficiently exploiting the available resources in cloud and edge environments, and therefore, facilitating the achievement of the performance requirements imposed on 5G.

On the one hand, SDN aims at mitigating the complexity inherent to the network devices involved in steering the multiple traffic flows of the network. To this purpose, SDN introduces programmable network elements so as to utilise this programmability to separate the control plane (*i.e.*, the decision-making plane responsible for electing the best suitable path for the multiple traffic flows) from the so called data plane (which refers to the actual tasks carried out by the network elements to redirect the traffic flows through the path specified by the control plane), centralizing all the logic in an upper hierarchy central node referred to as controller. In this context, the Open Networking Foundation (ONF) leads the definition of the SDN architecture [24], which aims at addressing three main aspects: *(i)* the communications between the controller and the programmable network devices; *(ii)* the exposure of the state information of these devices (and thus of the overall network) to external applications; and *(iii)* the interoperation among different controllers to overcome the potential issues of a centralized node such as reliability, scalability and security.

On the other hand, the NFV technology made its first breakthrough in 2012 with the stated purpose of tackling one of the most evident limitations in the networking arena: the dependency on proprietary hardware to provision the different functionalities that comprise a network. Due to the substantial importance of the NFV technology within the course of this thesis, the following section is dedicated to elaborating on the fundamental aspects of this technology.

2.2. Network Functions Virtualization Landscape

As previously commented, the last decade has witnessed the dawn of a new era where the virtualization and softwarization of network functions and components is expected to play a fundamental role in the provision of upcoming telecommunication services, as enabled by the integration of information technologies and networking. One of the key technologies to enable the paradigm shift is NFV.

The NFV technology made its first breakthrough in 2012 [25] with the stated purpose of tackling one of the most evident limitations in the networks arena: the dependency on proprietary hardware in the provision of network functionalities by telecommunication operators. This dependency entailed several notorious drawbacks such as a significant slowdown in upgrading the networks, ele-

vated investment and maintenance costs, and the barrier to new developers and manufacturers to promote innovation caused by requiring expertise in proprietary hardware. To overcome all these drawbacks, NFV relies on a virtualisation-based deployment of network functions, which allows to create an abstraction layer capable of decoupling the underlying hardware from these functionalities. The abstraction layer based on the network functions softwarization then provides the networks with a greenfield facility dedicated to: *(i)* deploy network services, *(ii)* reduce investment and maintenance costs (*i.e.,* related with operating and upgrading the network facilities), *(iii)* allow telecommunication operators to agilely integrate new network functionalities, reducing the network service creation time cycles, and *(iv)* to automatize the network management processes, preventing human error prone operations.

To make all this feasible, the ETSI has embodied the standardisation efforts of the NFV technology since its inception through a specific Industry Specification Group (ISG) for NFV (*i.e.,* the ETSI ISG NFV), bringing together a wide range of participants including network operators, manufacturers, different industry stakeholders, and research entities and universities. In this sense, the following subsections aim at presenting the most relevant aspects of the NFV technology, focusing on the basis defined by ETSI for the reference architectural framework. Finally, the section concludes with the review of the virtualization techniques that have supported the rapid adoption of NFV.

2.2.1. NFV architectural framework

As part of the previously mentioned standardization activities, the ETSI published the design principles of the NFV reference architectural framework [26], where it describes the elements that comprise the framework, and defines appropriate interfaces to enable interoperability among different vendor NFV-based implementations.

Following the ETSI view illustrated in Figure 2.2, the Virtualized Network Functions (VNFs) provide the traditional hardware-based network functions and service specific-functionalities (*e.g.,* firewall, DHCP server, IP router, residential gateway, *etc.*), implementing them in software. VNFs are interconnected to build end-to-end network services, which are defined as a composition of VNFs. Each composition denoting a network service is clearly defined by a Network Service Descriptor (NSD) that specifies: *(i)* the constituent VNFs of the network service, and *(ii)* the virtual links connecting these VNFs. Thus, VNFs may then be executed over a programmable substrate of hardware and software resources, which may be provided by server computers and other heterogeneous capacity equipment. These types of equipment conform to what is commonly known as the NFV Infrastructure (NFVI), and it supports the instantiation of VNFs with the utilization of virtualization technologies.

On the other hand, the Management & Orchestration (MANO) entity coordinates all the operations related to the lifecycle management of network services to be instantiated over the NFVIs. These include the deployment of the of VNFs comprising the service, encompassing their commission/decommission related to scaling operations, their interconnection, and the termination of the service (with the consequent destruction of VNFs). To this purpose, the MANO entity is further de-

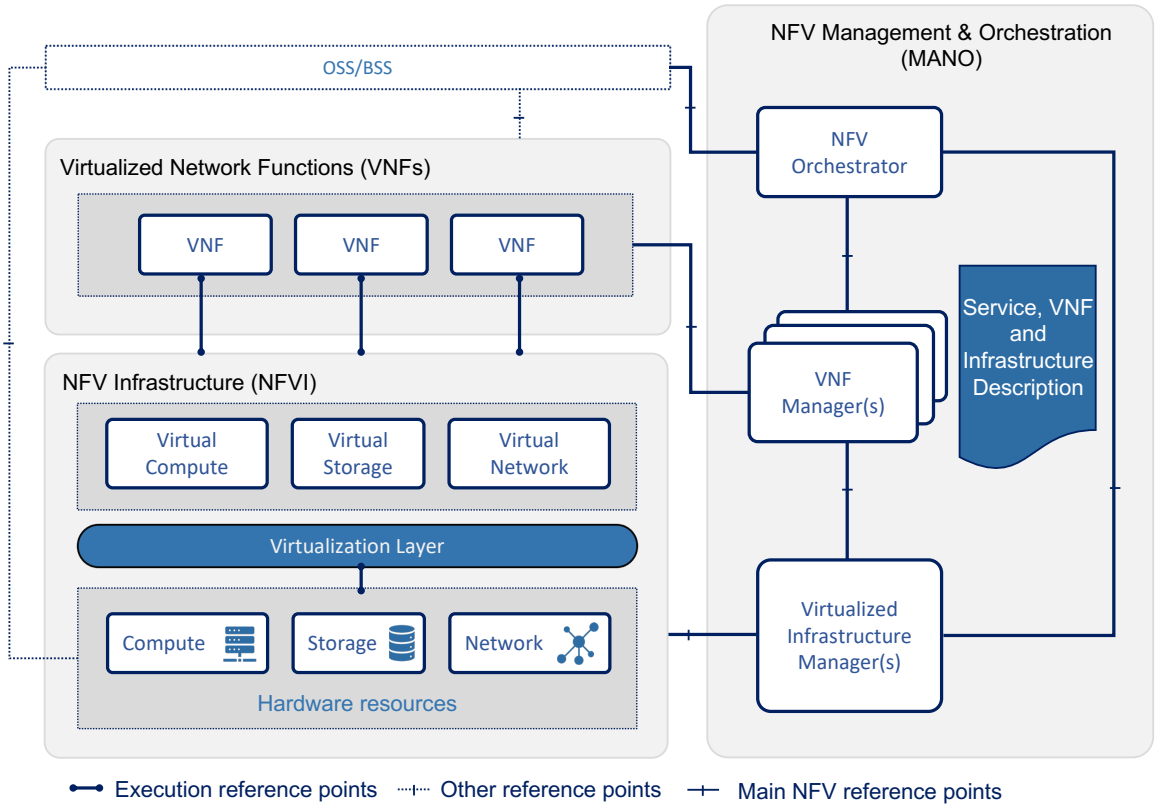


Figure 2.2. *ETSI NFV reference architectural framework.*

composed into three components within the ETSI reference architectural framework: the Virtualized Infrastructure Manager (VIM), which provides the functions needed to allocate, reallocate and scale the NFVI compute, storage and network resources to running VNFs, as well as to monitor the NFVI status; the VNF Manager (VNFM), in charge of managing the VNF lifecycle (*i.e.*, instantiation, configuration, modification, and termination); and the NFV Orchestrator (NFVO), whose role is to coordinate the allocation of resources interacting with multiple VIMs, as well as the lifecycle of network services by interfacing with different VNFM entities.

2.2.2. Management and orchestration solutions

As it is explained in the previous section, the MANO entity plays a fundamental role within the ETSI NFV reference architectural framework to carry out the coordination of all the operations related to the lifecycle management of the network service intended to be deployed. In this context, this section gives the reader an overview of the main MANO solutions available in the market, and also the selected one for the development of this thesis.

To this purpose, the main MANO implementations aligned with the ETSI architecture (reviewed in Section 2.2.1) were analyzed, considering the type of source licenses, the data models and languages used to describe NSes and VNFs, the ETSI–NFV architecture components included in their implemen-

Criterion / Solution	**ETSI OSM [27]**	**Cloudify [28]**	**Open Baton [29]**	**ONAP [30]**
License	Apache 2.0	Apache 2.0 for the free version (limited orchestration services)	Apache 2.0	Apache 2.0
ETSI NFV Components	NFV Orchestrator & VNF Manager	NFV Orchestrator & VNF Manager	NFV Orchestrator & VNF Manager	NFV Orchestrator & VNF Manager
Data model & descriptor language	YANG YAML	TOSCA YAML	TOSCA YAML	TOSCA YAML
Supported VIMs	OpenVIM [31], OpenStack [32], VMware Cloud Director [33], Amazon Web Services (AWS) [34]	OpenStack, VMware Cloud Director, Microsoft Azure) [35], etc. (it can be extended using plugins)	OpenStack (possible to extend it via VIM driver)	OpenStack (multi-cloud project will extend this support)
Supported VNFMs	Juju charms [36] (see Section 3.2)	Implemented by Cloudify	Implemented by Open Baton	Juju charms
Maturity	High	High	Medium	Medium
Starting date	First release announced in 2016	Founded in 2012	Project started in 2015	Initial release in 2017
Release Cycles	Every 6 months	Followed a two-year cycle since release 3.0.0	Not fixed	Twice per year (typically, every 6 months)
Status (up-to-date)	Under development. Current version: OSM Release TEN	Under development. Current version: Release 6.0.0	With no maintenance in its repositories since 2019	Under development. Current version: ONAP Honolulu

Table 2.2. *Management & Orchestration solutions.*

tation, the supported Virtualized Infrastructure Managers (VIMs) and VNF Managers (VNFMs), and the software development maturity of each solution. For this latter, the analysis has been extended to include also how such solutions have progressed over time since their inception up to the present. The results of this analysis are summarized in Table 2.2.

Based on this initial study phase, and as justified in Chapter 3, Open Source MANO (OSM) [27] was selected as the baseline technology to carry out the NFV-based research lines that are included in the following chapters of this thesis.

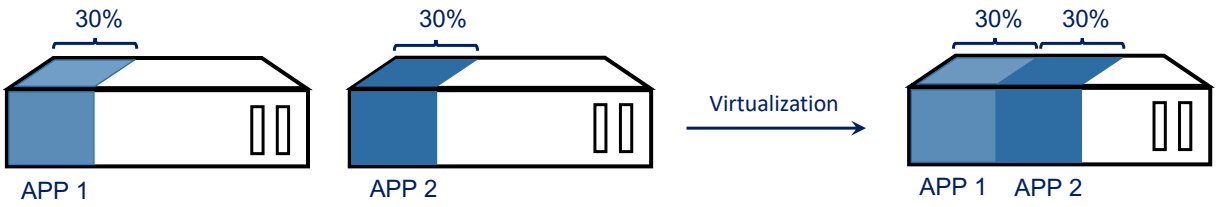


Figure 2.3. *Virtualization approach.*

2.2.3. Virtualization techniques supporting the NFV adoption

In the 1960s, the term virtualization was initially introduced to refer to the abstraction of dedicated physical resources or hardware for performing different computing activities. Originally, this approach emerged to address the challenging situation of the computer systems of that decade, which were only capable of processing one specific computational assignment at a time. Thus, the virtualization aimed at partitioning the resources of the same hardware computer server to allocate multiple computing tasks. Figure 2.3 illustrates this primary objective of the virtualization technology. However, the strong irruption of cost-effective machines with x86 microprocessors slowed down the development of the virtualization technology due to the more profitable approach of having different server computers, each in charge of a specific computing task, compared with the arduous process of having a single mainframe dividing the resources for different computing tasks.

With the technological evolution in the 2000s of the above-mentioned type of computers, the virtualization technology became relevant again since the use of the existing highly efficient hardware for a single application was a tremendous drain on resources, space, energy and cost. In this context, the virtualization moved further into the creation of software-based (or virtual) compute, storage, networking, servers or applications. This process is feasible due to the hypervisors, which provide the software layer that runs on the physical servers or hosts to realize the virtualization of the hardware equipment, pooling the resources from the physical server and allocate them to the virtual environments. There are two main types of hypervisors: *(i)* the so-called Type 1, or bare metal hypervisors, that are directly installed on top of the physical server. This class of hypervisors stands in place of a host Operating System (OS) and schedules the virtual resources directly on the hardware; and *(ii)* the Type 2, or hosted hypervisor, which involves an additional layer with the host OS that resides between the physical server and the hypervisor. Both hypervisors allow to building virtual environments or Virtual Machines (VMs). A virtual machine is basically a software-based computer that runs like a physical computer, with its own operating system (commonly referred to as guest OS) in charge of executing different applications. Accordingly, multiple virtual machines can be executed independently of one another on the same hypervisor since this latter manages the resources that are allocated to these virtual machines from the physical server.

In view of the above considerations, the virtualization technology introduced the following set of potential benefits into the computing arena:

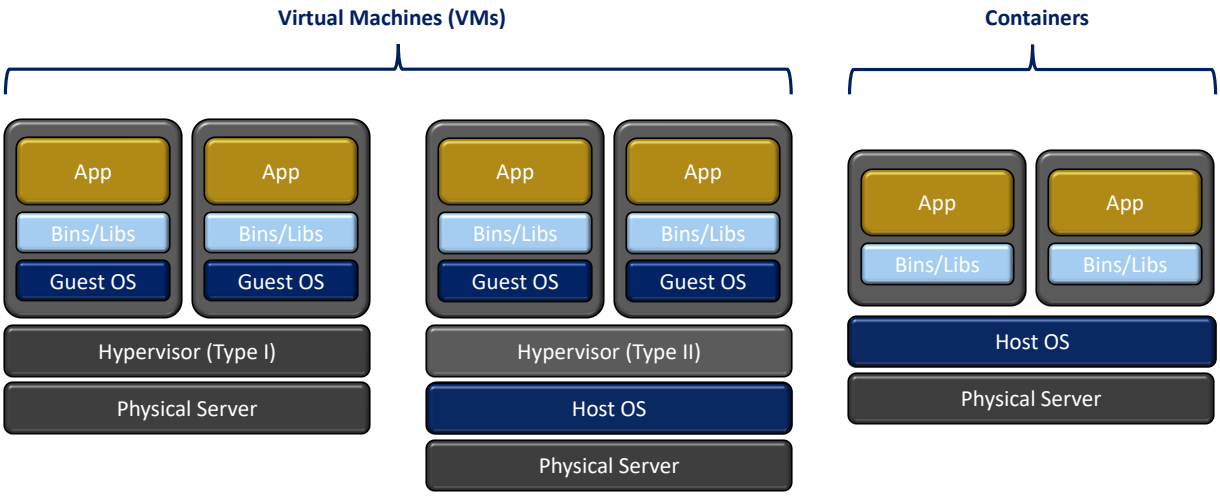


Figure 2.4. *Architectural principles of virtualization alternatives.*

- The ability of having multiple virtual environments within one physical equipment results in savings in investment and maintenance costs due to the drastic reduction of the physical infrastructure footprint.

- The flexibility and agility to create new virtual environments within a tighter period of time compared to the hardware appliances alternative (e.g., development scenarios to validate innovative solutions leading to their incorporation into production environments).

- The capacity offered by the virtualization to dynamically move or migrate the virtual machines from one hypervisor to another, on a different physical server, allows to implement redundancy strategies with the aim of decrease (or almost eliminates completely) the service downtime.

Notwithstanding the virtualization technology is a few decades old, it has undergone a constant evolution. Particularly, this technology branched out in 2008 towards a new virtualization technique defined as containerization. This innovative route in the virtualization emerged with the aim of supporting an abstraction of the application layer, bundling together all the essential elements (*i.e.*, configuration files, libraries and dependencies) that enable the execution of an application. Contrary to the classic hypervisor view, which aimed at providing an abstraction of the hardware layer, this containerization technique generally offers a lighter virtualization option since it does not require an OS specifically assigned to enable its execution. Thus, containerization provides the possibility of running isolated processes in individual virtual environments defined as containers, which share a common operating system. This capability allows containerization to not simply be a competing alternative in the virtualization arena [37], but also to integrate containers with VMs, enabling the software-based units encompassed by the containers to be executed within the virtual environment created for a VM. Figure 2.4 illustrates the architectural particularities of containerization, as well as those introduced by both types of hypervisors.

2.3. Service Orchestration over Resource-constrained Devices

Nowadays, virtual infrastructure management can be considered a mature and consolidated technology, with multiple open source and proprietary solutions in the field of cloud services. Examples of successful, widely adopted solutions are OpenStack [32], OpenVIM [31], VMWare vCloud Director [33], and Amazon Web Services Elastic Compute Cloud [34]. These solutions are typically used by cloud-computing service providers to support the execution of virtual functions through hypervisor-based virtualization technologies (reviewed in Section 2.2.3), supported by datacenters with high-profile server computers interconnected by high-speed local networks. As an alternative to traditional hypervisor-based approaches, container virtualization aims at providing a more lightweight solution, by sharing the OS of a compute node by all the containerized functions running on top of it. This approach enables to extend the application scope of the VIM towards more resource-constrained environments, where traditional hypervisor-based virtualization may impose inconvenient overheads in terms of computing and storage. Well-known examples of container-based technologies are LXC/LXD [38], Docker [39], or the leading orchestration platform, Kubernetes [40].

However, despite their reduced footprint in terms of computing, storage and networking requirements, these solutions still present limited applicability in highly resource-constrained environments and/or intermittently available, such as those that may be available beyond the access network segments of telecommunication operators. This is one of the factors that has led to the proliferation of initiatives to support the use of cloud native technologies in edge environments, such as KubeEdge [41] and OpenYurt [42], which extend the Kubernetes usage model to edge environments; or K3s [43], which offers a lightweight version of Kubernetes (with an estimated 50% memory savings, and size of less than 100 MB) suitable for remote environments with limited capacity. Other solutions based on fog computing principles are also worth mentioning. This is the case of Fog05 [44], an open source virtualisation solution that provides a decentralised platform for managing compute, storage and network resources in edge and fog environments. This solution could offer the potential to support resource-constrained distributed NFV environments, as observed in recent work in the field of virtual and augmented reality [45]. Table 2.3 collects a summary of different software projects and initiatives that provide virtual infrastructure management, with an indication on the capacity to support resource-constrained devices.

2.3.1. A particular resource platform: Unmanned Aerial Vehicles

The combination of 5G and Unmanned Aerial Vehicle (UAV) technologies is receiving increasing interest from the research community. Separately, they have an unquestionable impact on our society, with an increasing number of applications and value-added services exploiting them. It is only recently that new lines of research have started to be developed in order to explore the synergies between these two fields, considering aerial vehicles as enablers of 5G communications. As an example, authors in [46] proposes an architecture for the 5G networks, and for the next generations of mobile communication networks, that integrates UAVs. In this work, UAVs allow to accommodate occasional

Technology	**Brief description**	**Maturity**	**Functions**	**Resource-constrained support**
OpenStack	Open source software for creating public and private cloud platforms	High	VIM	✗
AWS	Comprehensive and broadly adopted cloud platform solution	High	VIM	✗
VMware	Cloud solution offering secure, flexible and efficient computing resources	High	VIM	✗
OpenVIM	Reference VIM included within the ETSI-hosted Open SourceMANO project	Medium	VIM	✗
LXC/LXD	Open source management extension for Linux containers	High	Containers orchestration	✓
Kubernetes	Open source system that facilitates the orchestration of containerized applications	High	Containers orchestration	✓
K3s	Lightweight Kubernetes distribution for resource-constrained appliances	At early stage	Containers orchestration	✓
KubeEdge	Open source system for extending cloud native containerized application orchestration to the Edge	At early stage	Containers orchestration	✓
OpenYurt	Open source computing platform intended to extend the cloud native ecosystem to edge computing and IoT environments	At early stage	Containers orchestration	✓
Eclipse fog05	Decentralized infrastructure for provisioning and managing compute, storage, communication and I/O resources	At early stage	VIM	✓

Table 2.3. *Summary of virtual infrastructure management software technologies.*

high traffic demands on the cellular access network, which may occur in specific geographical areas. In [47, 48], the research proposes to extend the coverage range of future wireless networks by deploying UAVs around a macro base station. These UAVs facilitate the relaying of user information and can also act as base stations to support end-to-end communications between multiple users. In a similar research line, [49], authors analyze the next generation of wireless communication technologies, identifying UAVs as key devices to reach the expected performance indicators for 5G networks. This is motivated by the ability of these devices to complement the resources of cellular radio access networks, and to facilitate network communications in remote areas.

On the other hand, as mentioned at the beginning of this section, the development of 5G networks does not only target to increase the performance of the communications offered to users. Taking

a more ambitious perspective, the aim is to build a new ecosystem in which technical innovations and business models respond to the use cases of vertical sectors. In this context, UAVs are currently gaining relevance as potential assets in different vertical sectors, such as precision agriculture [50, 51], smart cities [52, 53], logistics [54], and/or or public-safety [55, 56]. In these sectors, UAVs have been utilized as platforms for the generation, processing or transmission of relevant information (*e.g.*, video and sensor data generated in real time), as well as for the transport of objects and goods. The work in [53] presents a comprehensive study on the benefits, potential applications and challenges related to the use of UAVs to support wireless communication services.

With that noted, and due to recent advances in the miniaturisation of electronic devices, UAVs can on-board lightweight hardware platforms (*e.g.*, small computers powered by an external battery) with multiple wireless access interfaces (*e.g.*, based on line-of-sight radio links, Wi-Fi, or 5G/LTE cellular technologies). These platforms allow UAVs to be transformed into fully functional programmable compute nodes, which could be under the control of a VIM platform and support the execution of VNFs. Moreover, exploiting the intrinsic mobility capability of UAVs, these compute nodes could be easily positioned in specific geographic areas, even those that are extremely difficult to access. This enables the effective creation of on-demand NFV infrastructures. These infrastructures would allow the agile, flexible, and cost-effective provision not only of telecommunication services, but also of vertical sector services, in those geographic locations where such services are required. Thus, presenting a promising candidate for the practical realization of the vision outlined in this thesis, where UAVs offer a flexible, adaptable and programmable NFV infrastructure to support different telecommunications and vertical services beyond the network access segments of the telecommunications operators.

Focusing the background study of this section on the UAVs arena, [57] presents a detailed state of the art review of UAV-based communication networks, exposing the use of SDN to control aerial networks and facilitate the deployment and management of new applications and services. In [58], a concrete SDN-based solution for managing UAV aerial networks is presented. The solution introduces a platform for monitoring the aerial network, along with a traffic balancing algorithm to maintain an appropriate level of service in the network. The solution also considers the role of a UAV controller, which manages information related to the flight control, physical location and battery status of the UAVs. The UAV controller supplies information to the SDN controller, and can also accept commands from this latter in order to adjust the position of the UAVs, preserving the stability of the wireless communication links. The work in [59] also follows a holistic approach for the governance of UAV-based wireless ad hoc networks. The presented solution presents a centralised abstraction of the UAV network, which unifies network and flight control functionalities. This enables a flight operator to programmatically control the behaviour of the UAV aerial network, without the need to know specific details regarding the topology, the UAV mobility patterns, or the details regarding the control mechanisms of each UAV. The presented solution is based on SDN principles and software defined radio technologies. In [60], a system model is proposed to support the delay and computational requirements of novel vehicular applications (e.g., route planning, autonomous driving or infotainment services). The system is based on SDN technologies, edge computing and UAVs, and aims to optimize

the execution of computational tasks of these vehicular applications. To this purpose, an edge server and a UAV device are considered as computing platforms, and the UAV can also act as a relay node between the vehicles and the edge server. In [61], the authors introduce a video surveillance system based on UAVs for vast remote areas. In such a system, NFV enables the transmission of the video signal through a network of VNFs running on the aerial devices. Considering the resource limitations of UAVs, the authors propose the use of para-virtualisation. This allows virtual machines to share hardware resources, resulting in a more efficient management of available resources. The work in [62] considers UAVs as possible devices pertaining to a computing platform, capable of providing services through communications with other UAVs, and/or with other ground infrastructures. These services are provided through the use of middleware technologies. However, the authors do not refer to the limitations in computational capacity that these aerial devices typically present, nor do they consider the use of virtualisation or NFV.

CHAPTER 3

Open NFV MANO Platform for Multi-Site Experimentation

As the roll-out of the 5^{th} Generation of Mobile Networks (5G) has gained pace, automating the management and orchestration of resources needed for provisioning the forthcoming communication services and applications (*e.g.*, virtual reality, e-health, smart cities, etc.) has moved to the forefront as one of the essential building blocks for next generation networks. Particularly, this building block is included as a core element of the Network Functions Virtualization (NFV) technology defined by European Telecommunications Standards Institute (ETSI), which in turn is considered to be a key enabling technology for accommodating the postulated requirements such as higher data rates, ultra-reliable and low latency communications, reduced and optimized energy consumption, and ubiquitous network access, that those 5G novel services bring with them.

In this context, and with the aim of addressing the Objective O1 defined within the scope of this thesis, this chapter focuses on the design and deployment of the open ETSI NFV Management & Orchestration (MANO) platform of the 5G Telefonica Open Network Innovation Centre (5TONIC), the open research and innovation laboratory on 5G technologies founded by Telefonica and IMDEA Networks. This platform is intended to enable the creation of complex, close to reality, experimentation scenarios across a distributed set of NFV infrastructures, which can be made available by stakeholders at different geographic locations. For this, Section 3.1 motivates the creation of the previously mentioned NFV MANO platform, while Section 3.2 and Section 3.3 present the initial design principles and deployment aspects that were considered to provide 5TONIC trials and experiments with access to a functional production-like multi-site NFV environment. These aspects include the definition of an

inter-site communications model, the specification of a method that allow experimenters to securely access to the resources allocated within the NFV environment, and the exploration of novel mechanisms to support the configuration of virtual functionalities (*i.e.*, Virtualized Network Functions, or VNFs) deployed within the different NFV infrastructures integrated within the platform. Thus, enabling experimentation with novel NFV products and services. Section 3.4 presents the experimental activities performed to validate the deployed platform. Subsequently, Section 3.5 analyzes the progress and evolution of the platform since its first implementation, emphasizing how the design considerations detailed in the previous sections have enabled the development of different experimental activities and European projects. Finally, Section 3.6 closes this chapter, summarizing its most relevant contributions.

3.1. Introduction

To address the goal of pervasive, configurable, and highly reliable networks, the 5^{th} Generation of Mobile Networks (5G) comprises a wide range of technologies, with specific emphasis on the virtualization and softwarization ones. There is common agreement in that these technologies (currently focused on Software Defined Networking, SDN, and Network Functions Virtualization, NFV) are essential to provide the elasticity required to satisfy 5G requirements. One of the key elements in the provision of virtualized and softwarized networks is the orchestration of resources and functions, especially the aspects related to the management of the network functions implemented as cloud applications (Virtualized Network Functions, VNFs). The NFV community has coined the acronym MANO (Management & Orchestration) to refer to these mechanisms.

Therefore, a MANO framework is one of the essential infrastructures a 5G experimental facility must provide, at the same level of radio equipment or cloud-like infrastructure. This framework has to provide a consistent and robust access not only to developers or experimenters dealing with the basic network functionality, but also to users from the different verticals (*i.e.*, the application areas expected to make a direct and immediate use of 5G features) willing to experiment with 5G advanced services and new applications. The end-to-end nature of 5G services demands the support for multi-site integration, requiring multi-domain collaboration, and therefore implies the incorporation of these MANO capabilities. Thus, in an open laboratory, the deployed MANO framework has to be open itself, and facilitate the experimentation with its own components as well.

In this context, the 5G Telefonica Open Network Innovation Centre (5TONIC) [63] is an open laboratory founded by Telefonica and IMDEA Networks, with the goal of promoting collaboration in the development of 5G technologies. Since its start, several organizations of different nature have joined 5TONIC, running a wide variety of 5G related experiments and demonstrators, from radio access to advanced applications enabled by novel Network Services (NSes). 5TONIC hosts collaborative experiments run by combinations of members and participating organizations, many of them part of collaborative projects within the 5G Infrastructure Public Private Partnership (5G-PPP). Furthermore, 5TONIC has been a key component in the pan-European distributed testbed being deployed by the

5GinFIRE project [64], a natural continuation of the SOFTFIRE [65] initiative, and directly collaborating with related infrastructures worldwide, such as FUTEBOL [66] and ORCA [67]. During the ICT-17-2018 European call for multi-site 5G technology demonstrators, which was specially oriented towards the creation of experimental 5G infrastructures, 5TONIC was selected to become part of 5GVINNI [68] and 5G-EVE [69] European research projects. Finally, the European Commission recognized 5TONIC as a digital innovation hub [70], reinforcing its importance as a key co-creation laboratory in 5G, and its strong scientific and industrial role.

Under the above considerations, this chapter presents the design, deployment and validation of the 5TONIC open MANO platform. The work started with the selection of an open-source MANO implementation able to satisfy the above requirements, continuing with the laboratory deployment, the vertical and network experimenter support, and the integration with other sites. It is important to note that the temporal context in which the work of this chapter is centered (*i.e.*, in 2017), the available 5G experimental facilities in Europe [71] did not have many open source initiatives aiming to implement a MANO platform. Due to this, an additional effort was needed to understand, by means of existent implementations at that time, the implications and challenges of such a platform in practical situations.

3.2. Initial Design of the NFV MANO Platform

In order to carry out the initial design of the 5TONIC NFV MANO platform that is the main component of this chapter, the work started with the analysis and selection of an open source MANO implementation able to satisfy the above requirements. In this sense, the main MANO implementations aligned with the European Telecommunications Standards Institute (ETSI) architecture (reviewed in Section 2.2.1) were analyzed, considering the type of source licenses, the data models and languages used to describe NSes and VNFs, the ETSI–NFV architecture components included in their implementation, the supported Virtualized Infrastructure Managers (VIMs) and VNF Managers (VNFMs), and the software development maturity of each solution. For this latter, the analysis has been extended to include also how such solutions have progressed over time since their inception up to the present. The results of this analysis are summarized in Table 2.2.

Based on this initial study phase, Open Source MANO (OSM) [27] was selected as the baseline technology to build the 5TONIC MANO platform. Being an ETSI-hosted project, OSM emerged with the aim of delivering a functional implementation of a MANO software stack, following the ETSI specifications defined in the NFV standard. According to the previously mentioned analysis, this implementation provided appropriate stability properties and, being endorsed by relevant stakeholders of the telecommunications market, presented favorable prospects of evolution, with a commitment to provide regular releases under an open source license. Moreover, the OSM software stack was designed to be compatible with diverse cloud computing solutions, enabling the deployment of VNFs at multiple sites, as long as they support a compliant VIM. These features made OSM a suitable candidate to satisfy the requirements imposed to the 5TONIC MANO platform, and were the main drivers

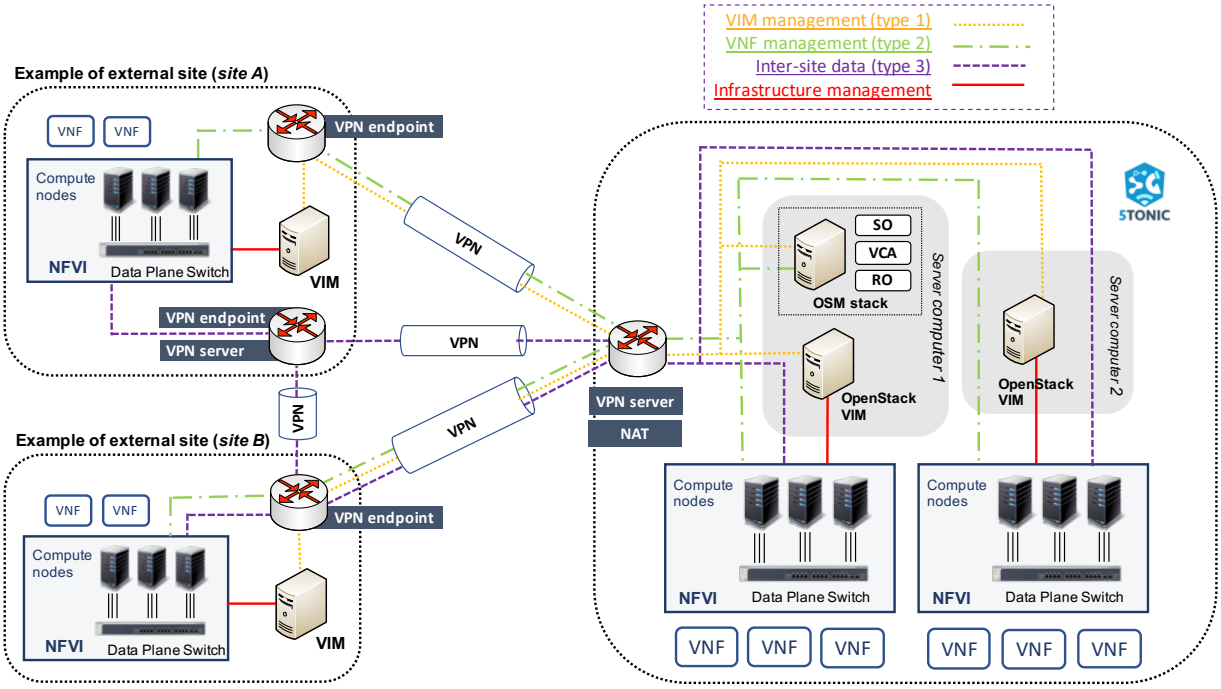


Figure 3.1. *Architectural design of the NFV MANO platform.*

to sustain its selection.

Figure 3.1 presents an overview of the architectural design of the MANO platform initially deployed at 5TONIC. At that initial time, the orchestration service of 5TONIC was based on the available release of OSM, *i.e.*, Release THREE [72]. In particular, this OSM release provided a Service Orchestrator (SO), a Resource Orchestrator (RO), and a VNF Configuration and Abstraction (VCA) module, components that are described next:

- The SO provides the point of contact for external entities (*e.g.*, the Operations Support System/Business Support System, OSS/BSS) to interact with the OSM system. It supports the lifecycle management of NSes, coordinating the creation and deletion of services composed of multiple VNFs, interacting with the RO and the VCA modules. Additionally, it provides other essential enabling functionalities, such as the management of NS/VNF descriptors and packages. In this sense, OSM Release THREE included a graphical user interface, offering an intuitive mechanism to ease the on-boarding of NS/VNF packages and the lifecycle management of NS instances.

- The VCA module represents the VNFM component defined by ETSI, supporting the initial configuration of VNFs after deployment (*i.e.*, day-1 configuration). With this purpose, the VCA uses the open source application-modelling tool Juju [36], which allows configuring VNFs through the execution of software scripts, referred to as Juju charms [36], that can be specified within VNF packages.

- The RO module coordinates the allocation and configuration of computing, storage and net-

work resources under the control of one or multiple VIMs, in order to support the execution and interconnection of VNFs. VIMs supported by OSM, via a plugin model, are indicated in Table 2.2. Additionally, this plugin model enables the RO to manage a number of SDN controllers, particularly OpenDaylight [73], Floodlight [74] and ONOS [75]. An analysis of relevant open source solutions regarding NFV and SDN can be found in [76].

In this initial design, the open experimentation environment of 5TONIC was structured into two independent local NFV Infrastructures (NFVIs), providing the hardware and software substrate to enable the appropriate execution of VNFs. Each NFVI was under the control of a VIM, supporting per NFVI isolation among experiments (for the implementation, OpenStack [32] was selected as the VIM solution). This organization of the MANO platform in two logically separated datacenters allowed the flexible allocation of experiments into independent infrastructures with different capacities. This configuration obeyed to the short-term needs identified for the projects and users carrying out experimentation activities at 5TONIC, and was specifically tailored to cope with future updates to accommodate changing requirements. To this purpose, the deployment of the MANO platform components relies on virtualization mechanisms. This allows to exploit the benefits of virtualization (*e.g.*, the possibility of realizing regular backups, or agilely migrating a component in case of failure on the hardware equipment) to carry out continuous integration techniques, and to allow the incorporation of any innovative features resulting from the evolution of the platform software base.

Besides this, the proposed architectural design enabled the flexible incorporation of additional sites and datacenters, with heterogeneous infrastructure and equipment as needed, to increase the portfolio of commodity and specific-purpose hardware available for NFV experimentation. Examples of these include internal or external datacenters operated by 5TONIC members, facilities from selected verticals, or external sites owned by partners of relevant research projects. The integration of these sites within the 5TONIC MANO platform was feasible through: *(i)* the multi-site capacity of OSM, which enables the automated deployment of NSes across multiple NFVIs, as long as they are under the control of compliant VIMs; and *(ii)* the enablement of effective and secure inter-site communications across the Internet and/or other non-trusted network domains. Both, the NFVIs of 5TONIC and the mechanisms for inter-site communications, are detailed in the next section.

3.3. Deployment of the NFV MANO Platform

This section elaborates on the most relevant deployment considerations of the MANO platform, carried out on the basis of the design principles presented above. Particularly, it includes the description of the main technical aspects of the experimental infrastructure hosting the MANO platform, the development of novel mechanisms that are integrated within the platform to allow both the inter-site communications and the access of experimenters to the platform, and an innovative contribution developed to support an alternative method to configure the VNFs that are managed by that MANO platform. It is worth noting that this deployment, which is totally based on existing open source tech-

nologies, was one of the first practical approaches to support vertical sector experiments in a multi-site NFV environment at a European level.

3.3.1. Description of the experimental infrastructure

Figure 3.1 illustrates the main NFVIs that were available for experimentation through the initial version of the 5TONIC MANO platform. As previously commented, the orchestrator software stack was based on OSM Release THREE. This software stack was installed in a virtual machine, which was executed in a server computer with four Gbps ports (*Server computer 1* in Figure 3.1). A second virtual machine in this server computer hosted an OpenStack release Ocata VIM [32], which allowed the allocation of experiments to a specific NFVI. This NFVI included three high-profile servers, each following a Non-Uniform Memory Access (NUMA) architecture, equipped with eight 10 Gbps Ethernet optical transceivers supporting Single Root I/O Virtualization (SR-IOV) capabilities. These were interconnected by a 24-port 10 Gbps Ethernet switch, used for data-plane communications. Two networks interconnected *Server computer 1* to the local NFVI: *(i)* an infrastructure management network, to support the management of local computing, storage and networking resources by the VIM; and *(ii)* a VNF management network, to enable the configuration of local VNFs via Juju.

The initial NFV experimentation platform at 5TONIC also included a second local datacenter, which could also be used to support the deployment of NSes through the multi-site support of OSM. This second datacenter included an OpenStack Ocata VIM, deployed as a virtual machine in a server computer with four Gbps ports (*Server computer 2* in Figure 3.1). This second datacenter offered an NFVI conformed by three server computers with the same hardware characteristics as the equipment hosting the VIM. Internet-access was provided to all components (*i.e.*, the OSM stack, VIMs and VNFs) through an access gateway and a Network Address Translation (NAT) function that is deployed within the 5TONIC infrastructure.

Given that the origin of this experimental infrastructure took place under the context of the 5Gin-FIRE project in the year 2017, it has undergone different transformations in relation to both the technical specifications and the configuration of the several elements comprising the platform. In any case, so as not to hamper the readability of this section with too many technical aspects, the current details about the platform have been included in Appendix A. Additionally, Section 3.5 later outlines the historical events that led to the current stage of the platform included in that appendix.

3.3.2. Inter-site communications

The following lines describes the main mechanisms adopted at 5TONIC to support inter-site communications. These mechanisms are necessary to enable the OSM stack at 5TONIC to coordinate the deployment and operation of network services at sites that are external to 5TONIC, each providing an NFVI under the control of a compliant VIM. From the point of view of the MANO platform, inter-site communications encompass the following types of data exchanges:

1) Communications between the OSM stack, deployed at 5TONIC, and the VIMs and SDN controllers operated by external sites. This type of communications allows the OSM stack to coordinate the allocation and configuration of computing, storage and network resources at the diverse datacenters.

2) Communications between the OSM stack and the VNFs deployed at each site, to support day-1/day-2 configuration of VNFs via Juju charms.

3) Inter-site communications between VNFs, to enable a VNF at one site to exchange data with VNFs deployed at other sites

The first two types involve the exchange of control information, and will be referred to as inter-site control-plane communications. The third type supports the exchange of data among VNFs located at different sites, and will be denominated inter-site data-plane communications. To enable these types of communications with external sites, the approach taken at 5TONIC consists in the utilization of an overlay network architecture based on Virtual Private Networks (VPNs). This approach is schematized in Figure 3.1, where the three types of communications are represented (type 1 referred to as *VIM management*; type 2 denominated *VNF management*; type 3 referred to as *inter-site data*).

It is worth mentioning that this overlay network approach requires a careful design of the IP addresses space to be used within the MANO platform, as well as by the external sites to connect to it, to enable effective network communications among those multiple site. For instance, to allocate the addresses space for inter-site communications without colliding with the actual addresses space already in use at every other site (reserved for other different purposes) is an essential condition. This led to the definition of an elaborated protocol to support the flexible incorporation of external 5G experimentation infrastructures, regardless their geographical distribution, in the 5TONIC MANO platform through the VPN-based overlay network architecture, and a validation method to verify the effectiveness of the integration. To facilitate the reproducibility of the protocol, assisting interested readers in configuring an NFV setup that integrates remote infrastructures, all the details were published in an open access video article [77].

As depicted in the figure, the VPN service that enables inter-site control-plane communications is deployed at 5TONIC. The VPN server offers authorized partners (*i.e.*, external sites providing experimentation infrastructures) a secure access with certificate-based authentication to 5TONIC premises, enabling the exchange of control-plane information. This way, inter-site control-plane communications follow a hub-and-spoke distribution model, where information is distributed using a star topology centered at 5TONIC (where the OSM stack is hosted). An infrastructure provider may determine the number of needed VPN endpoints (subject to agreement with the 5TONIC network operations center), and their location inside the provider site. This is shown in Figure 3.1, where two sites are connected to the VPN service hosted by 5TONIC. In this example, site *B* shares a VPN connection for inter-site control- and data-plane communications, while site *A* splits them into two independent VPN endpoints.

With respect to the exchange of data among VNFs deployed at different sites, a feasible approach consists in re-utilizing the aforementioned hub-and-spoke distribution scheme. In this case, the VPN server would act as a traffic relay among external sites. The 5TONIC laboratory is connected through a high-speed network access to the Internet and the GÉANT pan-European network. In addition, the VPN server is a high-profile server computer capable of handling received traffic at line rate. With these considerations, the network infrastructure that supports the inter-site data communications of the MANO platform is appropriate to parallelly run diverse demanding experiments. In this design, direct site-to-site communications are considered as an alternative in those virtual links with delay constraints, to ensure minimal end-to-end latency in specific experiments. This direct communications can be provided at the cost of provisioning additional VPN services at the sites involved in the experiment. Again, this is shown in Figure 3.1, where the site *A* deploys a VPN server, enabling direct communications with the site *B* to support the direct exchange of data among VNFs.

3.3.3. Provision of access to experimenters

Among the specific objectives that led to the creation of the 5TONIC MANO platform, a strategic aspect that was considered was to provide a multi-user NFV experimentation environment, *i.e.*, open for experimentation activities to authorized parties. In this context, OSM incorporated a role-based access control to provide permissions to users to selected projects defined by system administrators: users may have access to different projects and with different roles in each project, whereas projects provide the mechanism to group the access of users to resources. In addition, the role assignment allowed the project administrators to grant access to any or all the MANO services of OSM (*e.g.*, upload, modify and/or delete NSes and/or VNFs catalogues, allow users to manage the life-cycle of the NSes, etc.). With this, the administrator of the 5TONIC can handle the multi-tenancy of the MANO platform by adding/deleting projects/users and, depending on the existing agreements with verticals and experimenters, configure user accounts, assigning them specific permissions regarding the aforementioned services.

On the other hand, the execution of experiments imposes a connectivity requirement: after the deployment of NSes, experimenters must be able to access their resources under experimentation for administration, monitoring, reporting, etc., independently of the site in which their VNFs are actually deployed. In this respect, at the sight of Figure 3.1, the *VNF management network* (depicted in Figure 3.1 using the green colour) offered a suitable candidate to support the access of experimenters to their VNFs, as this network is mandatory for any new incorporated site and it reaches every deployed VNF. Simplicity is also granted in this solution, if centralized access to this management network is considered. In particular, the same certificate-based VPN mechanism provided by 5TONIC for inter-site communications was replicated for external experimenters, so that no additional complexity was added. From the security viewpoint, a single network shared by all external experimenters imposes the necessity of applying access control mechanisms to account for elements not being physically/-logically isolated. In that sense certain access rules are required to isolate control elements, blocking undesired flows and permitting only those strictly required.

In the context of the 5GinFIRE project, these security policies were implemented using *iptables* [78], which allowed to define the access filtering rules. These rules were then be applied in the OSM and VIM virtual machines at the logical interface attached to the *VNF management network*. In the case of OSM, the access filtering rules slightly depend on the the OSM Release, since those releases make use of different ports. For OSM Release THREE, the filtering rules look like presented below:

```
# OSM Release THREE iptables:

# Variable parameters in the rules are included between angle brackets
# (e.g., <interface_name>)

# Accept established traffic, i.e., responses to flows started at OSM.
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT -i <interface_name>

# Permit required IP addresses for full access (as many as required): e.g. 5TONIC operator.
sudo iptables -A INPUT -s <source_subnet> -j ACCEPT -i <interface_name>

# Permit required ports (specific for Release THREE):
sudo iptables -A INPUT -p tcp --dport 8000 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 4567 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 8008 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 9090 -m state --state NEW -j ACCEPT -i <interface_name>

# Permit required GUI ports
sudo iptables -A INPUT -p tcp --dport 8443 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT -i <interface_name>
sudo iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT -i <interface_name>

# Drop not required flows
sudo iptables -A INPUT -i <interface_name> -j DROP
sudo iptables -A FORWARD -i <interface_name> -j DROP
```

For the VIMs (*i.e.*, OpenStack release Ocata), the access filtering rules are presented next:

```
# OpenStack release Ocata iptables:

# Variable parameters in the rules are included between angle brackets
# (e.g., <interface_name>)

# Accept established traffic, i.e., responses to traffic started from VIM
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT -i <interface_name>

# Accept traffic from OSM
sudo iptables -A INPUT -s <OSM_source_IP> -j ACCEPT -i <interface_name>

# Permit required IP addresses for full access (as many as required): e.g., 5TONIC operator
sudo iptables -A INPUT -s <source_subnet> -j ACCEPT -i <interface_name>

# Drop not required flows
sudo iptables -A INPUT -i <interface_name> -j DROP
sudo iptables -A FORWARD -i <interface_name> -j DROP
```

In addition, the MANO platform included an access control gateway, denominated as Jump Machine, and controlled by the 5TONIC network operations center. As depicted in Figure 3.2, experimenters were enabled to connect to the VPN server of 5TONIC, being only granted access to this

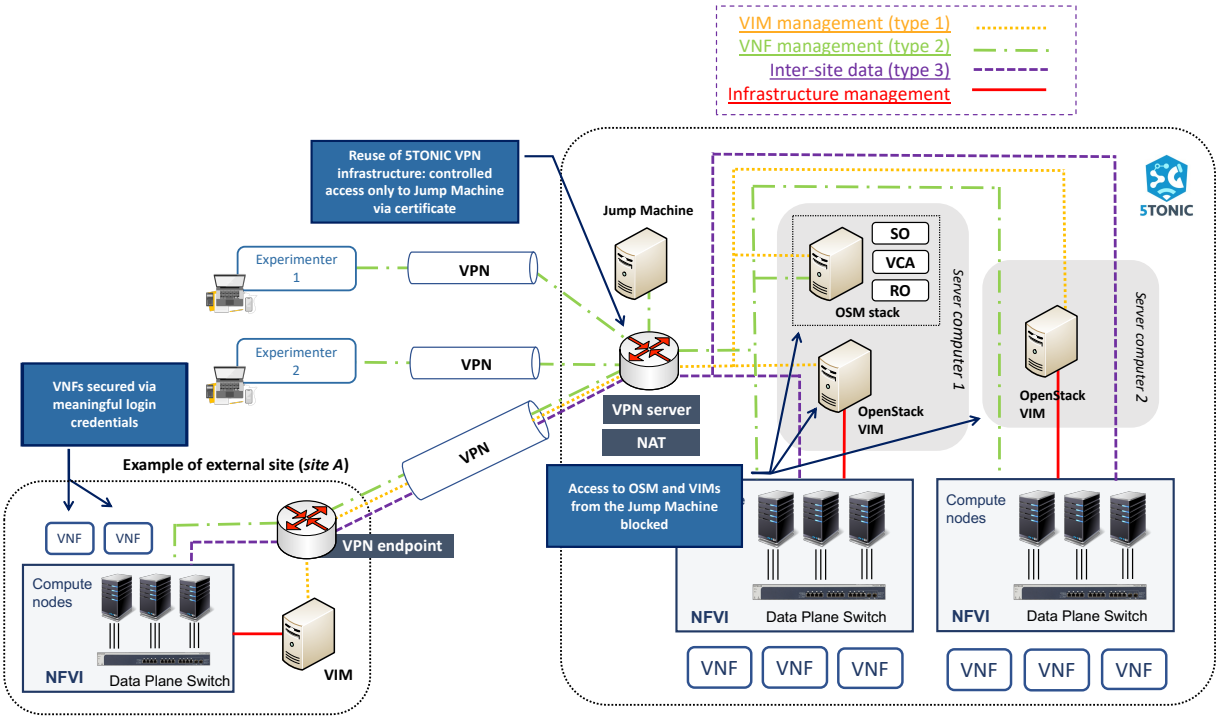


Figure 3.2. *Experimenters external access.*

gateway, and from there to any range of IP addresses (of all the external interconnected testbeds, not only from 5TONIC) that experimenters were allowed to connect. Specifically, the Jump Machine was implemented using a commercial Juniper M7i router [79], providing access control profiles and powerful filtering policies. Thus, controlling incoming connections to the VNF management network, and ensuring security for the orchestration modules.

3.3.4. Mechanisms to support the configuration of VNFs

As already commented, the VCA module is the OSM component in charge of the configuration of VNFs, generally aligned with the VNF Manager entity of the ETSI NFV architectural framework. It presents an interface to Juju, allowing the configuration of VNFs through the execution of a limited form of Juju charms, called VNF Configuration charms or proxy charms [80]. Essentially, the proxy charms are sets of reactive scripts packaged within the VNF descriptor for deploying and operating software that handle the VNF configuration. These charms are structured in layers [81], where every layer provides a software component that can be reused, *e.g.*, by combining it with other layers to extend the functionality of the configuration artifact. In particular, to carry out the VNF configuration tasks, two particular layers are required: *(i)* the *basic* layer, required by all the Juju charms; and *(ii)* the *vnfproxy* layer [82], that was released to aid the development of proxy charms. All the details related to how OSM manages the configuration of VNFs are presented in [83].

With the aim of extending the range of configuration options available to VNF developers in OSM,

as well as facilitating the portability of existing VNF developments, the utilization of other well-known and wide-used mechanisms for VNF configuration were explored during the realization of this thesis. Specially, Ansible [84] was identified as a technology of particular interest due to its wide adoption within the open source community. Ansible provides freely, under the software license Apache 2.0, an automation tool with a simple client–server architecture to handle the configuration and administration of one or several computing machines (either physical, or virtual). To this purpose, this technology includes a mechanism to easily describe the operations involved in the configuration of a computing machine through a particular plain, user-friendly language based on the YAML format, that is written in text files denominated *playbooks*.

Under the aforementioned considerations, this thesis encompassed the development of a novel base charm layer that allows to configure VNFs using Ansible *playbooks*. This base charm layer, named *ansible-charm* [85], extends the existing *basic* and *vnfproxy* Juju layers mentioned above, providing a template ready for customization to create a proxy charm capable of executing an Ansible *playbook* through the Juju framework included in OSM. Thus, this base charm layer provides OSM an alternative mechanism based on Ansible *playbooks* to carry out the configuration of the VNFs.

To utilize this development, the first procedure is to generate the proxy charm from the template provided, including the *playbook* with the intended configuration actions. This process is detailed step by step in the *ansible-charm* source code repository [85]. Once the proxy charm based on *ansible-charm* is generated, it has to be included in the directory structure that constitutes the VNF Descriptor (VNFD) (*i.e.*, the template that defines the properties of a VNF), following the information model defined by OSM for this type of descriptor [86]. Figure 3.3 shows the directory structure of an exemplary VNFD whose VNF configuration is based on the *ansible-charm*. After this, the VNFD can be integrated into a Network Service Descriptor (NSD), and be deployed with OSM.

The following lines summarize the operations sequence in which, once OSM initiates the instantiation process of a NS, the configuration is performed by means of the *ansible-charm*:

1) OSM proccesses the information included in the VNFDs, and identifies which VNFs of the network service requires configuration by the VNFM element in OSM (as indicated in Section 3.2, the VCA module implemented with Juju). At this stage, OSM commands Juju to start the preparation of the VNF configuration environment. To this purpose, Juju creates a Linux container (LXC) in the machine executing the OSM software stack for every VNF that requires configuration.

2) Once the configuration environment is prepared, Juju carries out the appropriate software installation within the LXC containers to perform the VNF configuration afterwards. This software installation is defined with the above mentioned layers included within the proxy charm. In this case, due to the *ansible-charm*, Juju installs an Ansible server in the LXC container. In addition, at this stage, Juju copies the *ansible-charm* directory included in the VNFD (see Figure 3.3) inside the LXC container, so that the *playbook* file is available for the Ansible server.

3) Simultaneously, OSM collects the access information that would allow the configuration of the VNFs. In particular, In particular, it gathers the management IP address of every VNF (provided by the VIM designate to allocate its resources), and the login information (*i.e.*, username and password) that are indicated in the VNFD.

4) Then, OSM transmits the access information to Juju so that it can complete the configuration of each LXC container. Since the container executes an Ansible server, this configuration will allow that server to establish a Secure Shell (SSH) connection to the VNF under its control, and to execute the configuration operations specified in the *playbook*.

5) Finally, once the Ansible server finishes the execution of the configuration actions, Juju sends a message to OSM to inform that the configuration process has been completed. Therefore, the deployment of the VNF is considered complete.

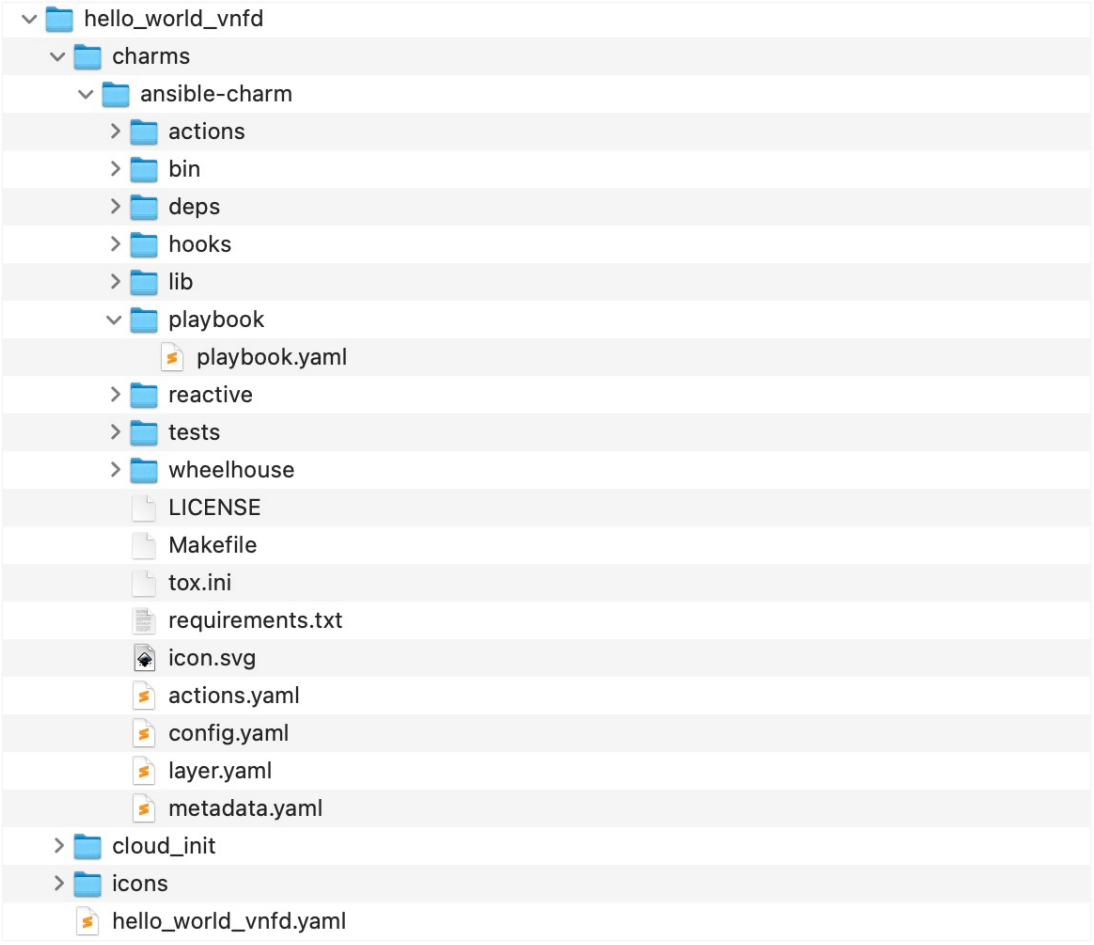


Figure 3.3. *Structure of a VNFD using ansible-charm.*

It is worth noting that, due to the high impact within the OSM community, and its wide-spread use due to the ease of describing the configuration instructions associated with a VNF, this development

was contributed to the OSM source code [87], being available with an open source license. In addition, this contribution was not only included in the release used throughout the research presented in this chapter (*i.e.*, OSM Release THREE), but has also been maintained throughout the successive OSM releases to date.

3.4. Practical Validation

This section addresses the set of experiments that were carried out to explore the performance of the 5TONIC MANO platform and its scalability properties. Firstly, a starting experiment was focused on studying how the deployment time of an NS could be affected by the number of its constituent VNFs. With this purpose, an NS was deployed at 5TONIC with a variable number of interconnected VNFs (1, 2, 4, 8 and 16). For each case under consideration, the deployment was repeated 30 times and calculated the average deployment time. The results of this experiment are shown in the left side of Figure 3.4. According to these results, the average time required to deploy an NS with a single VNF is approximately 54s. This time increases by less of 10s for each additional VNF that composes the NS.

In a second experiment, the aim was intended to evaluate how the deployment of an NS could be affected by existing deployments running at the same NFVI. With this purpose, 16 successive deployments of an NS with a single VNF were performed, measuring the time required for each deployment. Each cycle of 16 deployments was repeated 30 times. The right side of Figure 3.4 shows the average deployment time for the first, second, fourth, eighth and sixteenth deployment of the NS. According to the obtained results, it can be observed that the deployment time of an NS is not affected by previous deployments. Actually, it can be noted that the initial deployment is slightly longer because both the VIM and the compute nodes hosting the deployment do not have some deployment information (such as the VNF image) cached.

The results of both experiments indicate that the 5TONIC MANO platform had the potential to instantiate large NSes with appropriate deployment times, considering the key performance indicators established by the 5G-PPP [88] (*i.e.*, average service creation time not higher than 90 minutes).

Next, an additional experiment was defined to validate the ability of the 5TONIC MANO platform to accommodate multi-site NSes. In particular,this experiment involves the creation of an external site at Universidad Carlos III de Madrid (UC3M), consisting of an access gateway, an OpenStack release Ocata VIM, and an NFVI with two compute nodes interconnected by a Gbps switch. To enable inter-site communications (see Section 3.3.2) with the UC3M external site, the 5TONIC network operations provided a specific IP address space, along with the appropriate VPN credentials to gain authorized access to the 5TONIC MANO platform. With this, the external site was configured to use IP addresses within the allocated range, and to use the provided credentials to set up an access gateway as a VPN endpoint of the overlay network architecture. In addition, the experiment included the creation of virtual network at the external site, with layer-2 connectivity to the access gateway. Analogously, a virtual network was created at one of the 5TONIC datacenters, with network connectivity to the access gateway of the datacenter. Both virtual networks were interconnected via Virtual eXtensible Local

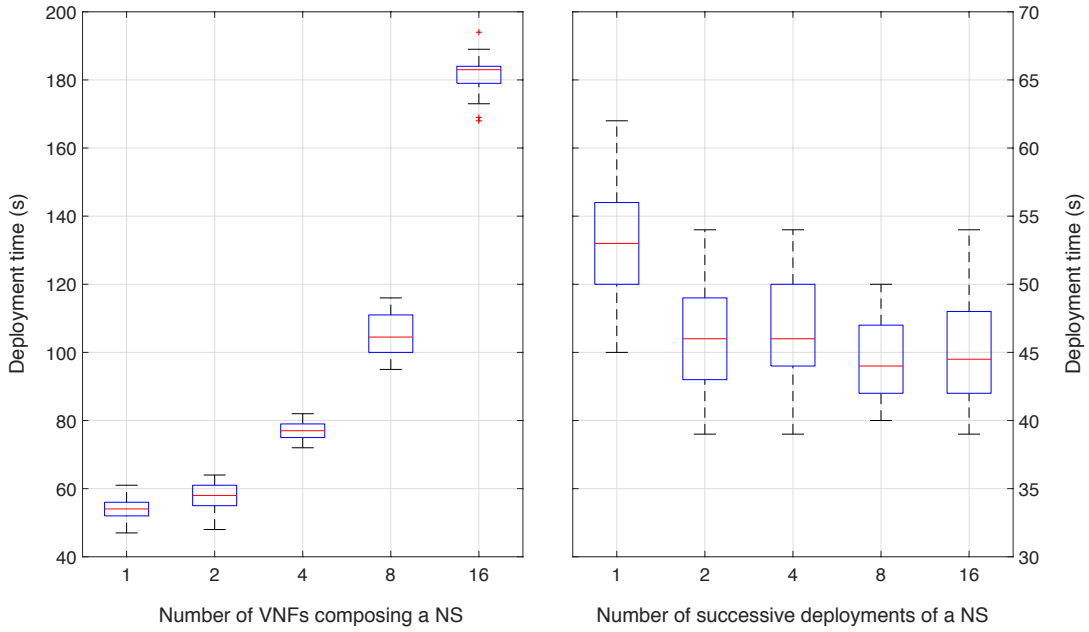


Figure 3.4. *Deployment times provided by the NFV MANO platform.*

Area Network (VXLAN) [89], configuring the access gateways as VXLAN tunnel end points.

Once the external site was ready and properly integrated within the platform, the graphical user interface of OSM was used to on-board the NS shown in Figure 3.5, as well as its constituent VNFs. The NS includes an Hypertext Transfer Protocol (HTTP) server function, which maintains a number of video files that can be requested on-demand by interested users using the HTTP protocol [90] (*i.e.*, a common approach used by existing video-on-demand services such as YouTube). A router function supporting the exchange of HTTP traffic between the server and remote network locations where video requests are originated. One of such locations is represented by a residential gateway function, allowing a user terminal to: *(i)* obtain IP access connectivity; and *(ii)* request a video file from the HTTP server, which is then streamed to the user terminal. Additionally, the residential gateway hosts a Dynamic Host Configuration Protocol (DHCP) [91] server that supports the automatic network configuration of the user equipment. The NS and the VNFs comprised within this experiment were made available under an open source license [92].

Likewise, the graphical user interface of OSM was used to instantiate the NS, allowing to carry out the deployment of the VNFs at different sites (*i.e.*, perform a multi-site deployment). In particular, the HTTP server and the router function were deployed at 5TONIC, while the residential gateway function was deployed at UC3M. Both the router and the residential gateway were automatically interconnected through the VXLAN, which enabled inter-site data communications. Day-1 configurations of VNFs (configuration of static IP addresses, activation of IP forwarding in the router and the residential

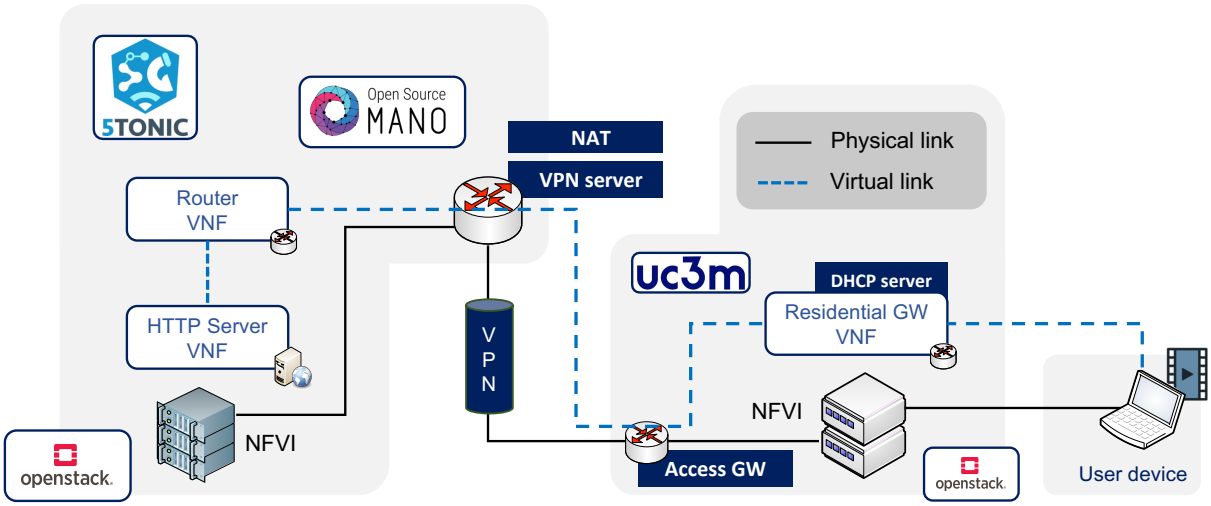


Figure 3.5. *Multi-site NS used for functional validation.*

gateway, and start of HTTP and DHCP services) were executed with Ansible *playbooks*, using the base charm layer *ansible-charm* presented in Section 3.3.4. After the deployment and configuration of the VNFs, a user device (a commodity laptop) was connected to the NFVI of the external site, obtaining access to the virtual data network of the residential gateway. After the laptop's interaction with the DHCP server, and completion of its network configuration, the software GStreamer [93] was used to request and play a specific video file from the HTTP server.

Figure 3.6 represents the Transmission Control Protocol (TCP) [94] throughput of the video delivery corresponding to the experiment, measured at the user equipment. The video file was received with an average rate of 4.44 Mbits/s, being uninterruptedly played out as it was received from the HTTP server. An analogous execution of the service, using a physical machine to deploy the HTTP server, and connected to the user equipment through a switch, provided similar throughput (also shown in Figure 3.6). This demonstrates that the performance of a service such as the one outlined above is not affected by being carried out in a virtualized form. In addition, it also validates the proper operations of the inter-site communications model presented, supporting the exchange of data among VNFs located at different sites.

3.5. Evolution & Progress of the MANO Platform

As previously commented, the technical solution adopted for the open multi-site NFV MANO platform of 5TONIC considered the utilization of a single NFV orchestrator, implemented using the ETSI-hosted OSM software. This is the element in charge of managing and coordinating the lifecycle of NSes. These services may be built as a composition of VNFs, which may be deployed at any of the sites integrated within the NFV ecosystem. The above detailed design of the 5TONIC NFV MANO platform (see Section 3.2) started in the context of the H2020 5GINFIRE project, where the platform was used to support the execution of more than 25 experiments, selected through a competitive open-

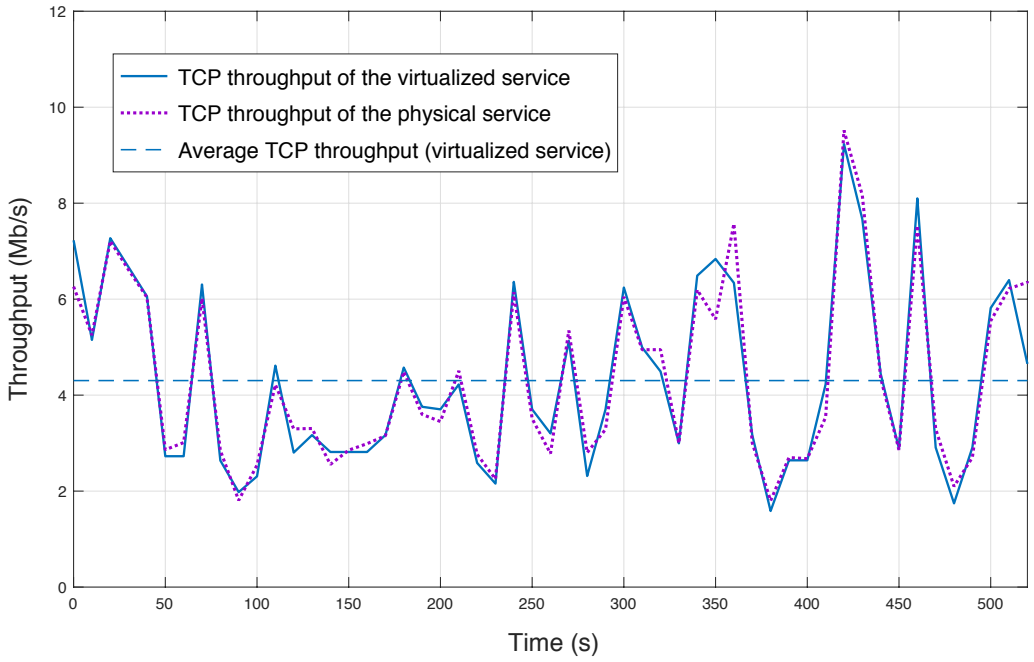


Figure 3.6. *TCP throughput measured during the experiment.*

call process, across eight vertical-specific experimental infrastructures located in Europe, and one in Brazil (this latter connected through a transoceanic link).

In addition, this platform embraced the integration of a singular NFV infrastructure in which the role of nodes providing computation is played by Unmanned Aerial Vehicles (UAVs), or drones (as they are most commonly known). This represents a major contributions of this thesis, and due to this, Chapter 4 is dedicated to elaborate all the details on this regard.

Likewise, the platform and the procedures defined on this chapter (*i.e.,* inter-site communications, provision access to experimenters, and mechanisms to support VNFs configuration) were leveraged to build a distributed NFV testbed at a national scale, in Spain. Particularly, this testbed was built spanning three different (remote) sites: Universidad Carlos III de Madrid (UC3M), Universidad Politécnica de Cataluña (UPC), and Universidad del País Vasco (UPV/EHU). The primary goal of this testbed was focused on exploring the synergies among NFV, UAVs, and vertical services from a practical perspective, and it supported experimentation activities within the Spanish 5GCity project [95]. All the details related to this testbed are further described in Chapter 5.

Subsequently, an additional Brazilian site was integrated into the platform, to support joint demonstration activities in the context of a research and innovation cooperation established between Brazil and Europe, *i.e.*, the 5GRANGE project [96]. More recently, the infrastructure was used to support third-party experiments in the scope of the 5G-VINNI project [68]. Last but not least, the presented

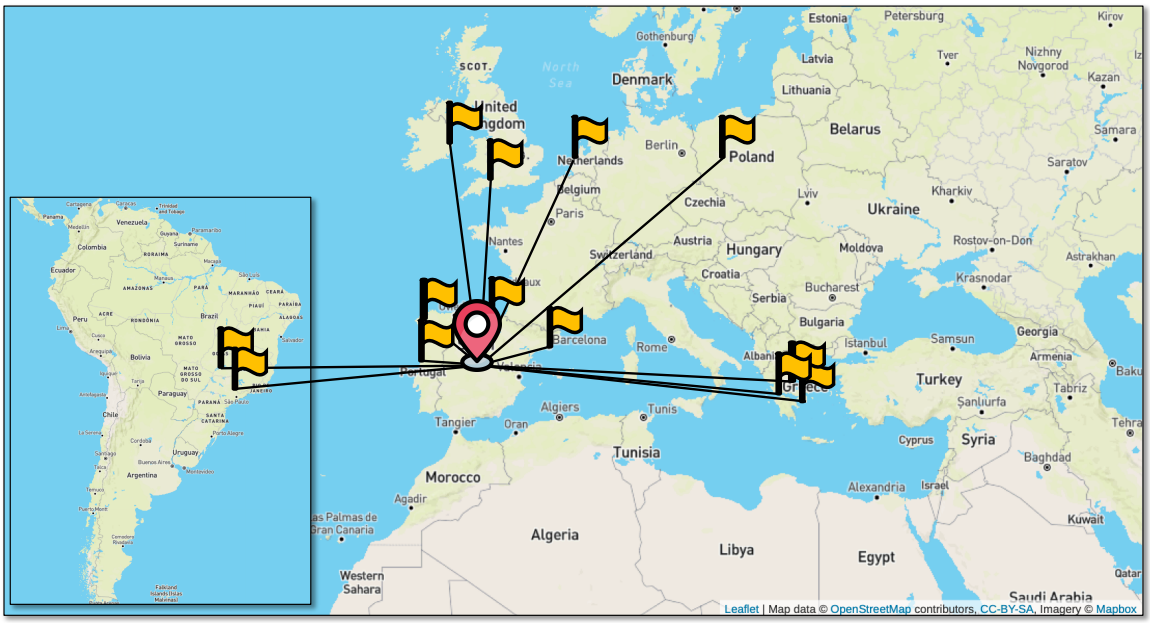


Figure 3.7. *Site distribution of the NFV MANO platform.*

NFV MANO platform has been evolved to support the work of UC3M under the context of two additional European H2020 research projects: FISHY [97], and 5GZORRO [98]. For this latter, the 5TONIC MANO platform accommodates diverse components of the project framework. Finally, it will be used as the basis for supporting joint research activities within the Spanish TRUE5G project [99].

The compendium of remote sites, along with their geographical distribution, that have formed part of the multi-site NFV MANO platform presented throughout this chapter is illustrated in Figure 3.7. Moreover, Figure 3.8 summarizes the overall evolution of this platform over the last few years.

3.6. Conclusions

The work presented throughout this chapter related to the 5TONIC MANO platform, has converged in diverse technical challenges, derived from its multi-user and multi-site requirements. As a result of addressing these challenges, this chapter has presented the following main contributions:

- Design and deployment of an open NFV MANO platform based on open source technologies, capable of accommodating experiments from different vertical sectors within a multi-site NFV environment. As previously commented, this occurred in a context in which NFV was just starting to receive interest from the industry and research community, and there were only a few initiatives related to the implementation of the NFV standard.

- The definition of a novel inter-site communications communications model to enable the deployment and operation of networks services across the several sites encompassed by the plat-

form. This mechanisms consists in the utilization of an overlay network architecture based on VPNs.

- The provisioning of secure access to authorized parties (*i.e.*, platform administrators, and experimenters who have resources deployed across the platform) to enable experimentation within the multi-site NFV environment.
- Development of an innovative solution (based on open source technologies) to extend the range of configuration options available to VNF developers in the selected MANO software stack, *i.e.*, OSM. This solution was contributed to the OSM source code due to its high impact, and widespread use within the OSM community.
- The realization of different experiments with the platform in order to validate its capacity to accommodate the deployment of network services within the deployment time requisite defined for 5G (*i.e.*, under 90 minutes). In addition, the experimentation includes the deployment of a video on-demand service to validate the capacity of the platform to manage the lifecycle of multi-site services, as well as the proper operation of the inter-site communications model to support the exchange of data among VNFs located at different sites.

This list of contributions underlines the significant relevance of this chapter within this thesis since it constitutes a first approximation to accomplish the remaining objectives, addressed in the following chapters.

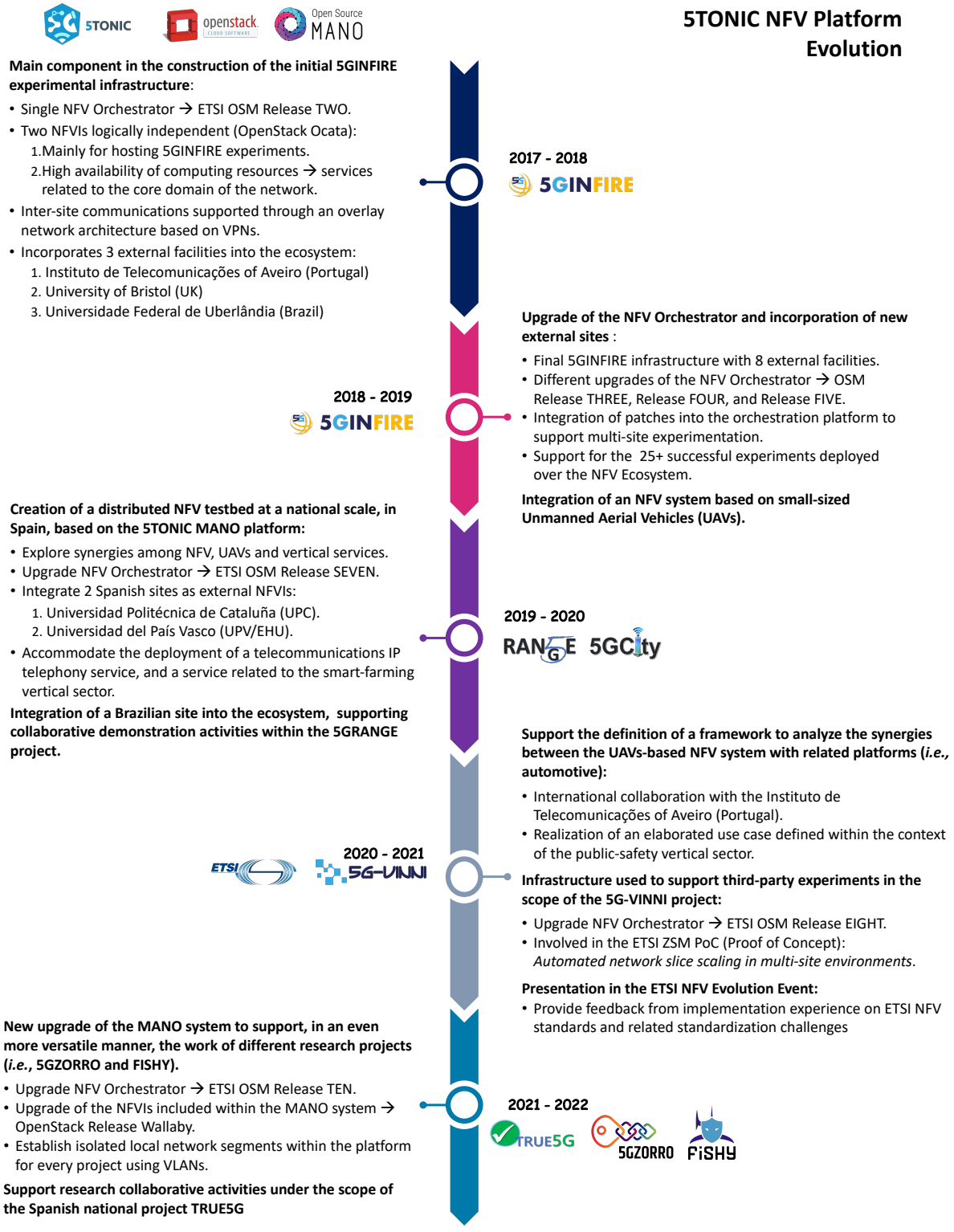


Figure 3.8. *Timeline of the NFV MANO platform evolution.*

CHAPTER 4

Adaptable and Automated UAVs Deployments via NFV

The continuous evolution of the Unmanned Aerial Vehicles (UAVs) in the recent years has opened new opportunities to support applications and services of great interest, not only for individuals, but also for society in general. In this sense, appealing applications and services are being considered by the industry and research community such as collaborative search and rescue, remote sensing in surveillance operations, or supporting telecommunications services in remote areas, to name a few. However, the conventional solutions of these aerial systems are commonly designed and manufactured to accomplish specific missions, and they can not be easily and dynamically reconfigured to adapt to changing mission objectives.

Building on the insights gathered in the previous chapter with respect to the Network Functions Virtualization (NFV) technology, and with the aim of tackling the Objective O2 defined within the scope of this thesis, the work of this chapter is focused on exploring the applicability of virtualization technologies and NFV standards into the UAVs arena. This approach will allow to constitute flexible UAVs deployments, adaptable to different requirements, and to support dynamic, cost-effective, and on-demand service provisioning beyond the network access segments of telecommunications operators. Thus, allowing the 5^{th} Generation of Mobile Networks (5G) to extend their programmable substrate, and provide reliable service operations in environments and situations where the primary telecommunications infrastructure may not be sufficient, nor available (*e.g,* due to an emergency situation, or in rural area environments). To this purpose, the main challenges and the potential benefits related with the utilization of NFV technologies in the UAVs environment, and the design of an NFV

system based on these aerial devices, are detailed in Section 4.2. Section 4.3 presents the prototype implementation details and the tests performed to validate the proposed system design, and finally, Section 4.4 concludes the chapter presenting the most relevant contributions and outcomes achieved under the context of this thesis.

4.1. Introduction

The progressive evolution witnessed by the Unmanned Aerial Vehicles (UAVs) during the last years, in conjunction with the services and products that can be offered by this kind of systems, have led to their consideration as an earnest candidate to support the development of new emergent applications and services. A deployment of UAVs typically consists of a Ground Control Station (GCS) that monitors and controls the operations of a set of Unmanned Aerial Vehicles (UAVs). These UAVs may have different capabilities (*e.g.*, through an heterogeneous set of on-boarded sensors), and can be deployed over a delimited geographical area to provide a particular service. In contrast to a usual use case where the services provided by UAVs are oriented towards the collection of data (*e.g.*, sensor values, video images, etc.), as well as their transmission to a ground station, the research community and the industry have recently considered the development of new value-added applications and vertical services where UAVs play a fundamental role. These services include remote sensing in surveillance operations [100], agribusiness [101], collaborative search and rescue [102–106], building aerial sensor networks to aid disaster management [107], or supporting backbone communications to mobile ground stations [108], to name a few.

On the other hand, the advent of the new generation of mobile networks (*i.e.*, the 5^{th} Generation of Mobile Networks, or 5G) has imposed new and stringent requisites to guarantee a fair balance between data rate, latency and cost of communications, which are expected to be potentially originated by millions of connected devices. For that purpose, the shared use of resources, allowing the dynamic provision of network functions, is foreseen as one of the key enablers of this new generation of networks. In this context, the Network Functions Virtualization (NFV) paradigm presents an innovative solution in the sector of information and communication technologies, aiming at supporting the active and high-performance processing of traffic delivered across 5G networks (more details already elaborated in Section 2.2).

Under the above considerations, this chapter explores, from a practical perspective, the applicability of NFV technologies to support the creation of a system based on UAVs to enable the cost-effective, and flexible vertical service deployments beyond the network access segments of telecommunications operators. In this approach, UAVs-based system offers a programmable NFV Infrastructure (NFVI) that enables the agile integration of services and functions, which may be determined by the operator of the UAVs at deployment time (*i.e.*, on-demand). Using Figure 4.1 as a reference example, UAVs could be deployed to provide a communications infrastructure over a disaster area, where networking facilities are insufficient or unavailable (*e.g.*, due to an earthquake or during firefighting activities). In this example, a set of UAVs could be used to deploy virtualized access points and rout-

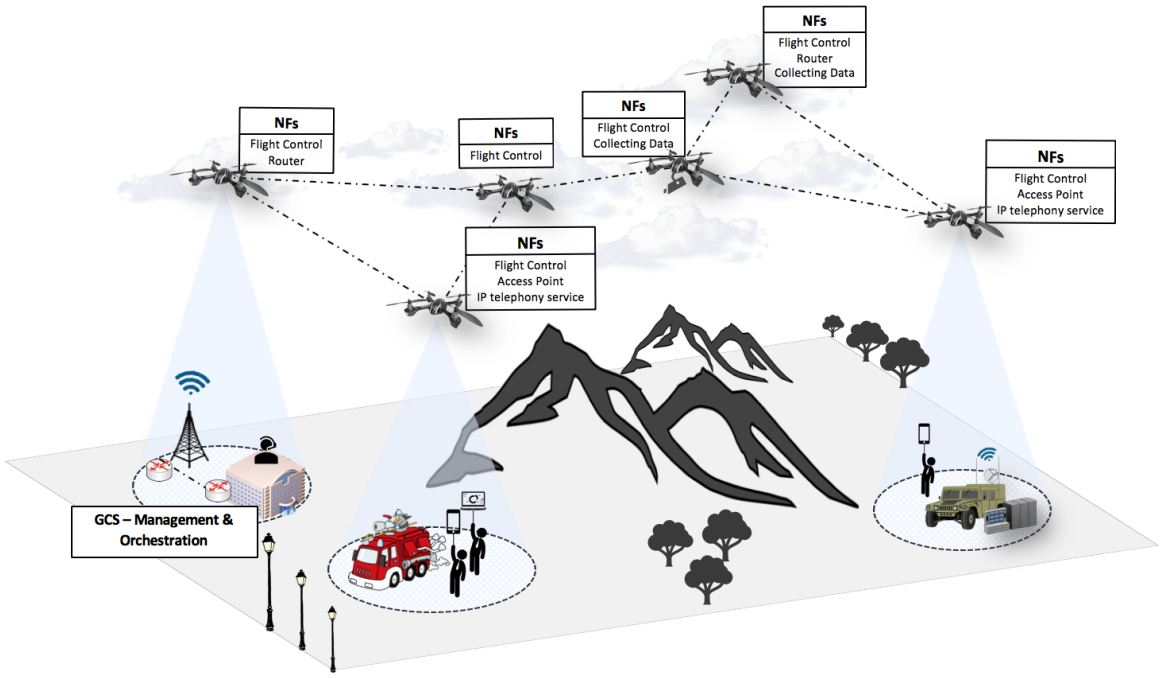


Figure 4.1. *Deployment of UAVs offering diverse network functionalities (NFs).*

ing functions, creating an aerial communication infrastructure over the deployment area. In addition, some UAV units could execute the diverse components of an IP telephony service, thus enabling real-time voice and video communications among the ground units operated by an emergency response team. Some aerial vehicles could also be configured to obtain and disseminate video images and temperature readings, improving the situational awareness and, consequently, the effectiveness of the adopted response actions. Furthermore, the same UAVs could later be used in a different mission, such as a search and rescue operation in a remote area. In this case, the flight control unit carried by the UAVs could be upgraded for certain aircrafts in the deployment, with the purpose of tracing different flight trajectories for an efficient inspection of a delimited search area.

Nonetheless, the full realization of this vision is challenging due, among other reasons, to: *(i)* the limited capacity of the hardware and software platforms that can typically be on-boarded on the UAVs; *(ii)* the need to automatically manage the resources provided by that platforms for deploying virtual functions on top of them, despite being transported by flying vehicles; *(iii)* the requirement to specify appropriate placement policies for virtual functions (*e.g.*, to indicate which virtual functions should be executed over the same UAV unit); and *(iv)* the energy consumption of the battery-powered UAVs, which is a limiting factor of the operation time. With the aim of addressing these challenges, this chapter presents the design of an NFV system capable of deploying network services over UAV platforms. In addition, it encompasses the technical details related to the implementation of the system prototype, and demonstrates its practical feasibility through the deployment of a realistic network service.

4.2. Motivation & System Design

As previously mentioned, this chapter studies the applicability of virtualization technologies and standards to build flexible UAV deployments, capable of extending the programmable substrate of 5G networks and therefore, enabling the service provisioning beyond the network access segments of telecommunications operators. With this objective, it considers the utilization of NFV, along with general purpose hardware and software platforms that can be on-boarded into small aerial vehicles. Thus, these platforms are able to provide the underlying substrate to execute network, transport and application level functions, which will be implemented in software and deployed over the UAVs as required using virtualization technologies.

The use of NFV technologies in UAV environments offers a number of potential advantages: *(i)* the ability to flexibly adapt the functions offered by UAVs to the specific mission requirements or objectives; *(ii)* the capacity to agilely deploy moderately complex UAV services and applications, installing and configuring the necessary components within a limited time period; *(iii)* it facilitates the development of new functions and services, with the possibility of testing them in controlled environments with similar characteristics compared to their corresponding production environments, as well as evolving and replacing them quickly; *(iv)* the provision of open platforms, which offers new possibilities for the incorporation of developers and manufacturers, including other communities in the sector of information technologies and communications to promote innovation; *(v)* the flexibility to accommodate deployment scenarios with changing resource demands, for example by incorporating additional virtualized functions or scaling them as it becomes necessary; and *(vi)* in general, reduction of investment and maintenance costs, through the use of general purpose platforms (*e.g.*, open platforms), whose manufacturing and supply costs can be reduced by economies of scale.

Despite the aforementioned potential benefits of using NFV in the context of UAVs, it also introduces a series of technical challenges that must be carefully considered. On the one hand, the hardware equipment that can be on-boarded in an UAV, providing the underlying substrate to support the execution of virtualized functions, is typically limited in terms of size and weight. This is especially evident in the case of the small-sized UAVs that have recently emerged in the market, which could at best carry a small set of Single-Board Computers (SBCs). In this way, the hardware platform supporting the execution of virtual functions present significant limitations in terms of computing capacity, networking and storage. On the other hand, such hardware must be integrated as part of the NFVI of an existing Virtualized Infrastructure Manager (VIM) solution, in such a way that it can be used by an NFV orchestration service to deploy virtualized functions. Additionally, it should be possible to perform an appropriate distribution (or placement) of Virtualized Network Functions (VNFs) to UAVs, in order to indicate which VNFs should be executed on the same aircraft. Moreover, the control communications that are necessary to manage the infrastructure resources carried by the UAVs should be maintained, even when these devices are flying. Finally, it must be considered that the UAVs have a battery, and therefore a limited operating time. For this reason, specific mechanisms may be required to support the replacement of these devices, including the migration of the virtualized functions hosted by them to new or existing UAV units. All these requirements are not common

in traditional virtualization platforms, where compute nodes are typically high performance servers, installed in a data-center, and interconnected through a high capacity fixed network technology (*e.g.*, an Ethernet network).

Given the complexity of these challenges, and aiming to guarantee the architectural and technological convergence of this work with existing solutions, the proposal was designed in accordance with recognized standards in the NFV arena. In particular, it considers the NFV reference architecture proposed by European Telecommunications Standards Institute (ETSI) [26], which is reviewed in Section 2.2. Figure 4.2 illustrates a conceptual vision of the proposed solution, serving as the basis for the definition of its corresponding architectural design. Taking into account the capacity limitations of the typical hardware platforms that can be on-boarded on UAVs, the VNFs should be developed to provide their intended functionality, but making a responsible use of the available resources. For this reason, in the vision illustrated in the figure, these VNFs are referred to as lightweight VNFs. In general terms, the solution design is structured into three main components:

1) **The Management & Orchestration (MANO) system**, located in the GCS, is the component in charge of the management and orchestration of available hardware and software resources, as well as the deployment and interconnection of the lightweight VNFs. It provides an orchestration service, with the execution of an NFV Orchestrator (NFVO), a VNF Manager (VNFM), and a VIM. The GCS offers a stable environment with appropriate resources for the operation of this component, which is convenient given its criticality.

2) **The NFVI**, encompassing the hardware and software infrastructure that supports the execution of VNFs. As already mentioned, this infrastructure comprises the UAVs as the computational units that can execute lightweight VNFs. Additionally, this infrastructure entails devices that can operate on the ground (labelled in Figure 4.2 as *Ground unit(s)*), providing additional computational capacity to the infrastructure, and supporting the execution of VNFs with a more resource demand.

3) **The Mission Planner**, also located at the GCS, which is the entity responsible for specifying the descriptors of the network services to be deployed, each one as a composition of VNFs, in addition to the configuration parameters of each VNF and the policies that determine the allocation of the VNFs to the UAVs. The mission planner generates this information in concordance with the mission requirements that are provided by the operator of the UAVs at deployment time.

It is important to note that this chapter is mainly concerned with the aspects related to the management and orchestration functionalities over the UAVs-based NFV system. For this reason, and given that the service offered by the Mission Planner is orthogonal to those functionalities, the chapter mainly focuses on the other two components of the design. Particularly, on the architectural design of the infrastructure components (*i.e.*, the UAV devices), considering the different challenges derived from their mobility and limited-capacity.

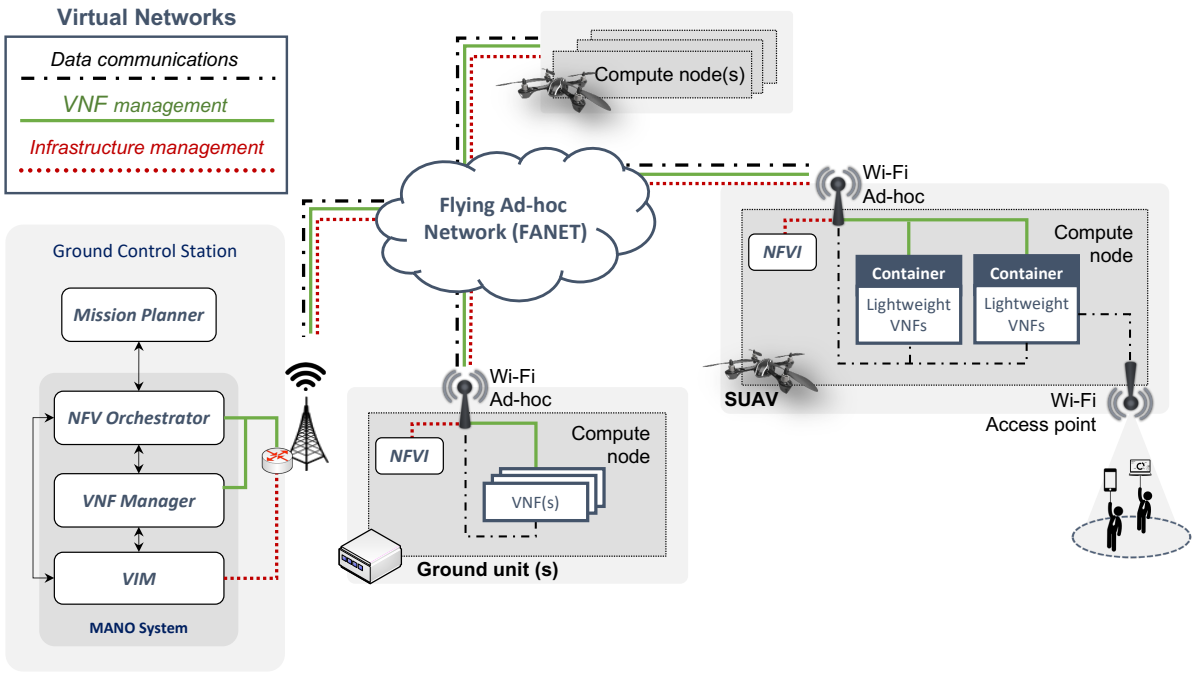


Figure 4.2. *Overview of the UAVs-based NFV platform design.*

Focusing on the architectural design of the UAV devices, each UAV carries a general purpose hardware and software platform. This platform, hereafter referred to as a compute node following cloud-computing terminology, has a wireless communication interface that enables the exchange of data with every other component of the design that is within the radio coverage of the compute node (*i.e.*, every other compute node, whether flying unit or not, and the MANO system). This wireless interface can be based on different technologies, *e.g.*, it may provide a line-of-sight or Wi-Fi radio link. Additionally, as reflected in the figure, some compute nodes may include a secondary network interface to provide wireless access connectivity to end-user devices.

In this design, the nodes (both the on-boarded at the different UAVs, and the located in the ground) form a wireless ad-hoc network that enables multi-hop data communications (that is, communications among compute nodes at different units, and between the compute nodes and the MANO system). These data communications across the ad-hoc network are supported with the execution of a Flying Ad-hoc Network (FANET) routing protocol at each compute node (*e.g.*, AODV [109] or OLSR [110]). In this sense, the design assumes that the support of this routing protocol is enabled at each node as a prior step to an operational deployment. Besides, to isolate the different traffic types exchanged over this wireless ad-hoc network, this solution defines a set of virtual networks that operate over the wireless ad-hoc network.

In particular, to support management operations towards the compute nodes (*e.g.*, to monitor the available resources at each compute node, to instantiate and configure a VNF instance, or to terminate that instance), each compute node included in the NFV infrastructure supports two types of communications: *(i)* communications between the VIM and the compute nodes (labeled as *Infrastructure*

management in Figure 4.2), to control the computing, storage and networking resources of the compute nodes; and *(ii)* communications between the NFV orchestration service and the VNFs (denoted as *VNF management* in Figure 4.2), to manage the configuration and the lifecycle of the VNFs. In this design, each of these types of management communications is delivered over an independent virtual network, which is created over the wireless ad-hoc network infrastructure offered by compute nodes (*i.e.*, the virtual network operates over the FANET routing protocol executed at compute node). Thus, management communications are isolated and delivered "in-band", through the aerial network infrastructure conformed by the compute nodes. Finally, given that management communications are always necessary for any application requiring the deployment of an unmanned aircraft system, both virtual networks can be pre-configured offline at each compute node and the GCS.

Regarding data communications among the VNFs that conform a network service (indicated as *Data communications* in Figure 4.2), these are also isolated using virtual networks that are built on top of the wireless ad-hoc network established by the compute nodes. The number and configuration of these virtual networks will typically be application-specific (*e.g.*, IP telephony traffic would require an independent virtual network). Hence, they will automatically be created by the MANO system as required, under the indication of the Mission Planner.

In an operational deployment, the VNFs of a network service (*e.g.*, the components of an IP telephony service) would be instantiated and configured at every involved compute node. In addition, the MANO system would create the virtual networks that are necessary to interconnect the VNFs and support data communications. In this solution, the flight control functionalities required at each UAV unit can be provided by a lightweight VNF. This lightweight VNF will execute the needed actions to position the UAV at its target position (the position, as well as the trajectory of the UAV to reach that position, can be configured at the flight control VNF by the MANO system). Management communications (*i.e.*, between the MANO system and each VNF, and between the VIM and each compute node) are still maintained through the virtual networks that have been pre-configured offline for this purpose, with the support provided by the FANET routing protocol that operates at the compute nodes.

Finally, as a specific design criterion regarding the deployment of virtual functions, it is important to note that, given the limited-capacity hardware and software platforms that can be provided and/or transported by UAVs, the solution is based on container virtualization technology, as opposed to traditional hypervisor-based virtual machines, to support the deployment of the lightweight VNFs. This is a crucial consideration that has a significant impact on the practical feasibility of the proposed design.

4.3. Validation of the Solution

This section describes the most relevant aspects regarding the validation of the previously presented solution design (see Section 4.2). In addition, a number of VNFs (including lightweight to be executed over the UAVs) were developed to support the execution of a moderately complex network service, which allowed to test and validate the proposed design. In this sense, both the system prototype and the VNFs were implemented using open source software technologies.

Figure 4.3. *UAV-based NFV system prototype.*

4.3.1. Prototype implementation

The use of open source technologies was considered one of the basic principles of the implementation process. In particular, Open Source MANO (OSM) [27] Release FOUR (available release at the time of conducting the implementation activities of this chapter) was selected to provide the functionalities corresponding to the orchestration and the lifecycle management of lightweight VNFs. With respect to the VIM, the implementation utilized the set of software tools provided by the well-known cloud computing platform OpenStack Ocata [32]. Both the OSM stack and the VIM provide the MANO system of the system design, and were installed over two virtual machines running in a mini-ITX computer (Intel Core i7 2.3 GHz, 16 GB RAM, 128 GB SSD, 4 GbE ports). This specific device offers the necessary computational capacity for the proper execution of both virtual machines, in addition to be compact-size, which makes it portable.

Regarding the UAV platforms, the system prototype includes a set of aerial vehicles Parrot Bebop 2 [111]. The selection of this UAV model was based on its ability to carry an SBC Raspberry Pi 3 Model B (RPi). These RPis are used as the compute nodes of the design, supplying the needed resources in terms of computing, storage and networking, and supporting the execution of the lightweight VNFs. The RPi boards are incorporated as compute nodes of the OpenStack VIM, through the necessary configurations to enable virtual networking through Linux networking bridges. Besides, each RPi includes an integrated Wi-Fi interface, and a number of them also contain a secondary wireless interface provided by a Wi-Fi USB adapter. The first interface enables air-to-air and air-to-ground ad-hoc communications with other UAVs, ground units, and with the MANO system. The second interface is intended for deploying a wireless access point, capable of providing network access connectivity to mobile end-user units (the suitability of these multi-interface devices to support multimedia communications was validated in a prior research work [112]). Along with the UAV platforms, the infrastructure also integrates a modular mini-ITX computer platform (Intel Core i7 2.3 GHz, 16 GB RAM,

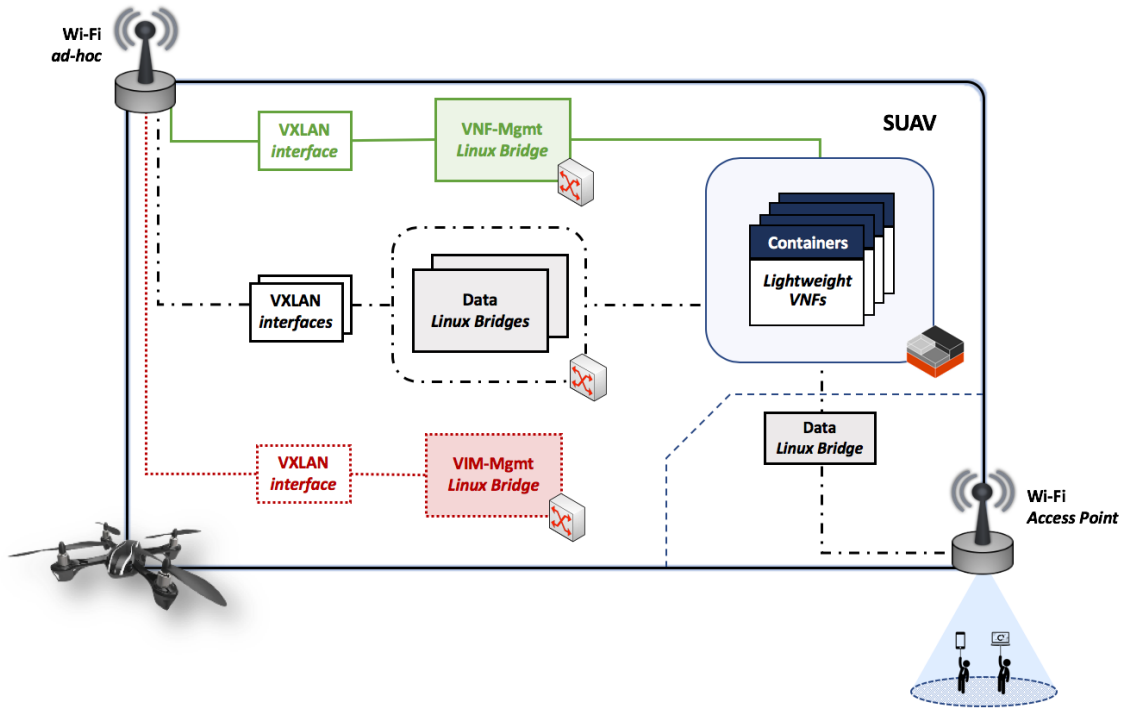


Figure 4.4. *UAV-based compute node network configuration.*

128 GB SSD, 4 GbE ports) with Linux Operating System (OS) as a compute node, complementing the infrastructure resources offered by the SBCs, and supporting the execution of VNFs that require more powerful equipment by means of the Kernel-based Virtual Machine (KVM) open source virtualization technology. Figure 4.3 shows these components that constitute the prototype of the NFV system based on UAVs.

With respect to the virtual networks that enable both control and data plane communications that take place in the platform, the standard Internet protocol Virtual eXtensible Local Area Networks (VXLAN) [89] was selected. The VXLAN protocol presents a feasible solution to exchange control and data traffic over a wireless ad-hoc network since: *(i)* the VXLAN traffic can be sent over the Wi-Fi interface in ad-hoc mode that is available at every compute node (directly sending traffic from a lightweight VNF, deployed over a virtualization container, through a Wi-Fi ad-hoc network is challenging, as this is not currently supported by the Linux kernel of the RPis); *(ii)* the VIM, as instructed by the OSM stack, can dynamically create virtual networks based on this protocol to interconnect lightweight VNFs hosted by different compute nodes; and *(iii)* the utilization of VXLANs does not require additional network configurations (*e.g.*, network routes) at the intermediate RPi boards that conform the network path between two communicating entities (*e.g.*, between two VNFs) due to the tunnelling mechanism supported by VXLAN.

Figure 4.4 details the network configuration that was done at each compute node. Firstly, a VXLAN interface attached to a Linux bridge (*VIM-Mgmt* in the figure) enables the control communications

required by the VIM to manage the computing, storage and networking resources available at every compute node. Secondly, an additional VXLAN interface allows the MANO system to carry out the life-cycle management of the lightweight VNFs hosted by the UAVs. In this case, the Linux bridge attached to the VXLAN interface (*VNF-Mgmt* in the figure) allows communications with every lightweight VNFs running at the same compute node. Note that both VXLANs are statically pre-configured at every compute node to enable the control communications of our system, as presented in Section 4.2. Finally, the configuration of the compute node allows the dynamic creation of virtual networks for data communications among VNFs (residing at the same, or at different compute nodes), as requested by the VIM and according to the instructions provided by the OSM stack.

4.3.2. Validation scenario

The experiment carried out to validate the potential of the UAV-based platform to execute realistic and moderately complex network services is described herein. The experiment encompasses the implementation of the multi-site Network Service (NS) illustrated in Figure 4.5, along with the elaboration of the NFV descriptors (*i.e.*, Network Service Descriptor (NSD), and VNF Descriptors (VNFDs)), and the development of every single VNF that constitutes the end-to-end service. Both the NFV descriptors, and all VNFs (including the lightweight VNFs) were developed and made available under an open source license[1,2].

In particular, the experiment offers a telecommunications service in which users in the vicinity of small-sized aerial vehicles may access to an IP telephony service provided within the facilities of a telecommunications operator. Although this experiment was focused on this specific service, this scenario may serve as reference to deploy other type of services like: *(i)* providing connectivity in a remote area location, beyond the coverage of cellular infrastructures; *(ii)* supporting communications in emergency operations such as fire extinction, search and rescue activity, etc.; and *(iii)* disseminating information from remote areas at programmed times of the day.

To realize the multi-site experiment depicted in Figure 4.5, the prototype implementation elaborated in Section 4.3.1 was built at Universidad Carlos III de Madrid (UC3M), and integrated as an external site into the 5TONIC NFV Platform described in the previous chapter (see Chapter 3).

With respect to the lightweight VNFs (*i.e.*, the VNFs that are executed over the resource-constrained UAVs) involved in the provision of the IP telephony service, the NS includes two virtual access points (*Router/AP* in Figure 4.5), which support data communications between end-user equipment (*e.g.*, IP phones) located at different areas. Each of these virtual access points holds a Dynamic Host Configuration Protocol [91] (DHCP) server, which automatically provides the required network configuration to allow IP access connectivity to connected users. In addition, the service comprises a lightweight VNF providing the functionality of a Domain Name System [113] (DNS) server (*Router/DNS* in Figure 4.5), resolving host names to IP addresses during the execution of the IP telephony service.

[1] NFV descriptors: https://github.com/5GinFIRE/mano/tree/master/descriptor-packages
[2] VNFs images: https://vm-images.netcom.it.uc3m.es/JoVE/

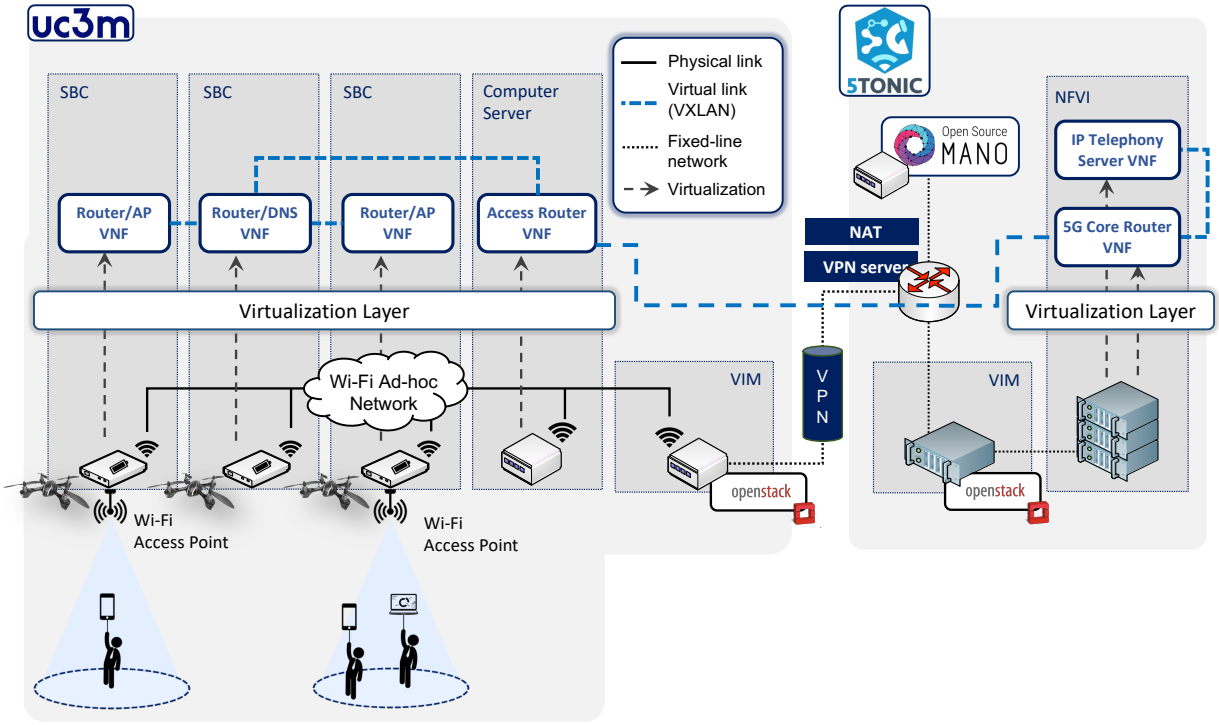


Figure 4.5. *Validation scenario.*

On the other hand, the IP telephony service enables the exchange of signalling traffic to establish multimedia sessions between communicating endpoints, *i.e.*, Voice over IP (VoIP) calls among user equipment at ground units. To support this service, a VNF providing the functionality of a VoIP server was developed based on the open source software Kamailio [114]. This VoIP server (labelled in Figure 4.5 as *IP Telephony Server*) enables the establishment of voice communications with the utilization of the Session Initiation Protocol (SIP) [115].

In addition, the service involves the development of a basic 5G core network defined by the 3rd Generation Partnership Project (3GPP), capable of providing connectivity to end-users in a secure manner, whether or not access comes from a 3GPP access network [116, 117]. In the case of non-3GPP access networks, the Non-3GPP Inter-Working Function (N3IWF) would be responsible for providing access to the core network, ensuring confidentiality, integrity and authentication in the course of communications. From this perspective, the provision of the basic forwarding functionalities defined by the 3GPP for the N3IWF element provides the implementation of a basic 5G core network prototype (*i.e.*, there are elements of the 5G core network defined by the 3GPP that need to be develop). To this purpose, the service includes two additional VNFs (labelled in Figure 4.5 as *5G Core router* and *Access router*). On the one hand, the *5G core router* implements the user-plane protocol stack defined by 3GPP for a N3IWF, and supports the network routing functionalities within the 5TONIC domain. Meanwhile, the *Access router* runs the user-plane protocol stack defined by 3GPP for a 3GPP User Equipment (UE) to reach the core network via an untrusted non-3GPP access. This VNF also supports the network routing functionalities to enable the connectivity of the VNFs accom-

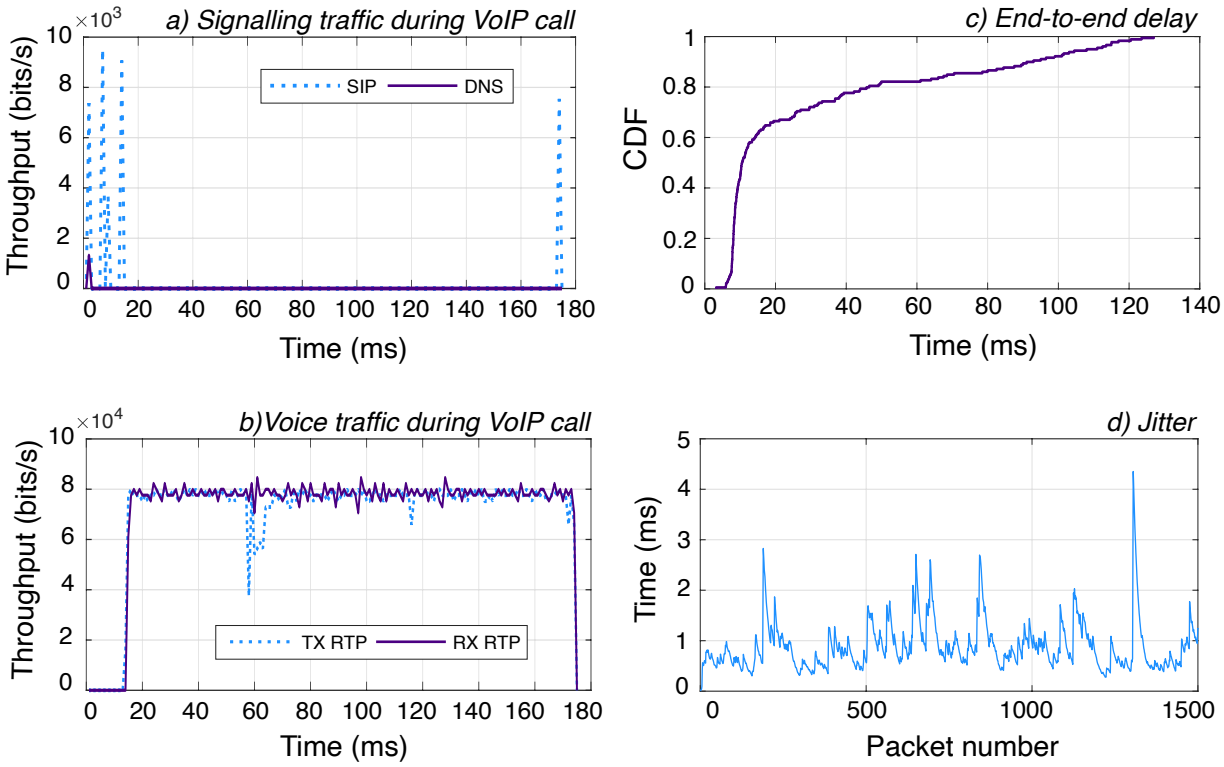


Figure 4.6. *Validation measurements.*

modated by the UAVs infrastructure with the core network or Internet. Finally, both VNFs make use of the Generic Routing Encapsulation (GRE) [118] and Internet Protocol Security (IPsec) [119] network level protocols with which the 3GPP stipulates this secure access. Thus, the implementation of both VNFs supports the user-plane stack defined by the 3GPP for non-3GPP access networks.

With this, the NS was instantiated using the client application of OSM. This instantiation results in the execution of the aforementioned VNFs in virtual containers at the UAVs, following the disposition indicated in Figure 4.5. For that end, the definition of each UAV as an availability zone (*i.e.*, a set of resources) in the OpenStack VIM allowed to execute each lightweight VNF at the expected UAV through the placement policies provided by OSM. The lightweight VNFs composing the IP telephony service communicate through a set of VXLANs that are dynamically established over the Wi-Fi ad-hoc network. These VXLANs are created on-demand by the VIMs, as instructed by the OSM stack. Besides, the configuration of all the VNFs after their deployment (commonly known as Day-1 configuration) is done through Ansible *playbooks* [84], using the developed open source base charm layer [85] (already described and outlined as one of the relevant contributions of this thesis in the previous chapter).

Once the deployment and the configuration of the VNFs composing the IP telephony service was completed, two wireless VoIP phone ZyXEL Prestige 2000W were connected to each of the access points offered by the NS with the objective of carrying out a VoIP call. Figure 4.5.a illustrates the traffic exchanged during the VoIP call established by the VoIP terminals, including the DNS and the SIP signalling messages needed to execute the call. The figure Figure 4.5.b reflects the voice traffic

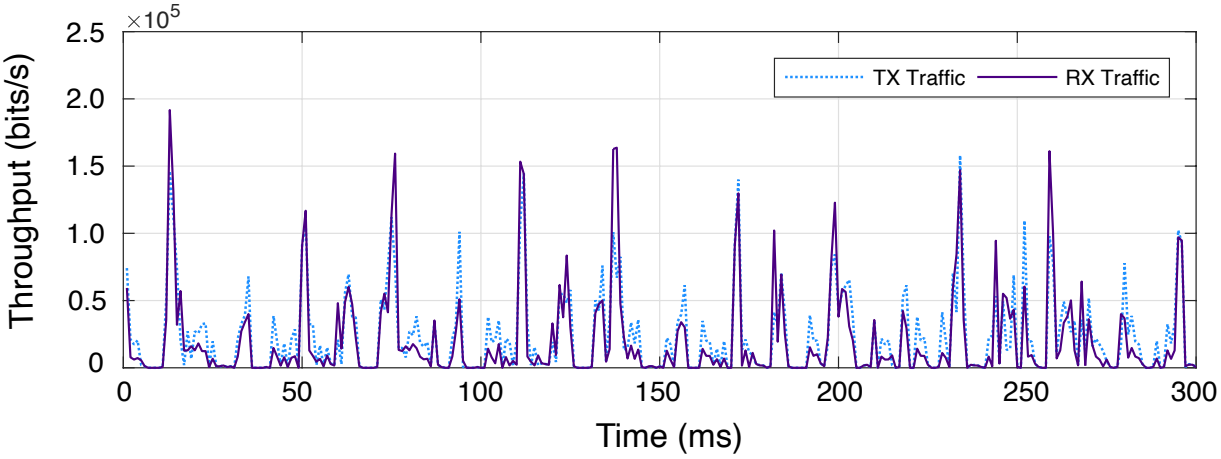


Figure 4.7. *NFVI control communications.*

transmitted and received by one of the wireless phones.

In addition, both jitter and end-to-end latency were measured with the aim of verifying the robustness of the provisioned deployment to support this type of services. As can be seen in the Figure 4.5.c, more than 80% of the end-to-end delay measurements were below 60 ms, and none of them were higher than 150 ms, which guarantees appropriate delay metrics for the execution of a voice call. With respect to the jitter, Figure 4.5.d shows the metrics collected in the forward direction with an average value lower than 1 ms. These results satisfy the recommendations specified by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [120] for supporting VoIP services. Accordingly, the voice call progressed with no glitches and good sound quality.

These experimental results validate that the proposed system, based on virtualization technologies and NFV standards, can effectively support the automated deployment of network services over the limited-capacity compute and network resources provided by the UAV platforms. In any case, it should be noted that these results have been obtained in a laboratory environment and with the UAV devices grounded or following a limited and well-defined flight plan. Thus, ensuring the seamless connectivity among all the components of the system, and therefore, enabling the proper operation of the virtual overlay networks. Other scenarios involving outdoor deployments may introduce environmental conditions, affecting the stability of the flight of the UAVs, and hence the performance of the IP telephony service. Due to this, further work is required to carefully analyze how the mobility of the aerial vehicles, and their consequent intermittent connectivity, could affect in the functioning of the system.

On the other hand, these results also suggest the potential of the proposed system to be used in other environments, or vertical sectors, where resource-constrained hardware platforms might be available with the needed capacity to execute virtualized functions (*e.g.,* in smart-cities, or in the connected industry sector). However, the applicability of this solution to different environments and its potential adaptations require a careful study in a case-by-case basis.

Finally, Figure 4.7 represents the control traffic generated and received periodically by the MANO system, which is exchanged in order to monitor and synchronize the information of the available/consumed resources at each compute node. This traffic was measured to get an insight on how control traffic operates in our platform once an NS is being executed, and to verify how this traffic may affect the provided service. As expected, the measurements show that this traffic is negligible (with an average of 20.65 kbps) compared with the traffic exchanged during the execution of an NS, *e.g.*, during the VoIP call. Therefore, these low traffic rates suggest that control communications do not negatively affect the operations of moderately complex network services deployed with the proposed system.

4.4. Conclusions

This chapter has contributed this thesis with a novel solution to support the flexible, automated and cost-effective deployment of network services over UAVs. In this sense, this solution relies on the virtualization technologies and NFV standards to enable the automated and adaptable provisioning of 5G services (from both telecommunication and vertical sectors) beyond the network access segments of telecommunications operators. In this context, this chapter has presented the following main contributions:

- Design of a system based on virtualization technologies and NFV standards to support the flexible and cost-effective of 5G services. To this purpose, the proposed system considers the UAVs to extend the 5G programmable network substrate, and presents the architectural design of these aerial devices to support the execution of lightweight VNFs (*i.e.*, VNFs with moderate resource demands).
- Definition of the main drivers to support both the control communications between the GCS (and therfore, the MANO stack communications) and the NFVI composed by the UAVs, and the data plane communications (*i.e.*, the communications exchanged by the VNFs that are executed by the aerial devices) by means of virtual overlay networks. These overlay networks are established over the network conformed by the UAVs and the ground units, leveraging FANET routing protocols to support IP communications that allow the correct operation of these overlay networks.
- Implementation of a functional prototype of the proposed system, based on open source technologies, and validation of its practical feasibility. For this latter, the experiments included in the validation experiments involved the deployment of a realistic telecommunications service. In particular, an IP telephony service.

In the following chapter, this thesis takes a step forward in this direction, and explores the potential of this system to support vertical services, considering a more comprehensive context: *(i)* on the one hand, with a national-scale experimentation infrastructure; and *(ii)* on the other hand, in collaboration with related limited-NFVIs (*e.g.*, automotive infrastructures).

CHAPTER 5

A Multi-Site NFV Testbed for Experimentation with SUAV-Based 5G Vertical Services

With the advent of 5G technologies, vertical markets have been placed as fundamental drivers and adopters of technical developments and new business models. Within these markets, Unmanned Aerial Vehicles (UAVs) are gaining traction as key assets to generate, process, and distribute relevant information for the provision of value-added services. However, the enormous potential of UAVs to support a flexible, rapid, and cost-effective deployment of telecomunications or vertical applications is still to be exploited.

Due to this, and with the aim of addressing the Objective O3 defined within the scope of this thesis, this chapter explores the utilization of the previously described Network Functions Virtualization (NFV) system based on UAVs in the context of different vertical environments. To this purpose, the work of this chapter includes the design and development of a multi-site NFV testbed at national scale. All the specificities of the multi-site NFV testbed design are described in Section 5.2, whereas Section 5.3 presents the validation of the UAVs-based NFV system, with a special emphasis on the support of multi-site UAV services, through the realization of a specific use case related to the smart-farming vertical. In addition. this section analyzes the results obtained with the validation experiments, particularly in terms of service deployment delays. Section 5.4 concludes the chapter with a discussion of the lessons learned, and summarizing its most relevant contributions.

5.1. Introduction

As previously commented in Section 2.3, the combination of 5^{th} Generation of Mobile Networks (5G) and Unmanned Aerial Vehicles (UAVs) has recently turned out to be more than just a trendy idea. Separately, they have an unquestionable impact in our society, and an increasing number of new related applications are continuously appearing. However, it has only been recently that the amalgamation of these fields has received attention from the research community, and their natural synergies are beginning to be explored considering aerial vehicles as enablers of 5G communications [46], [47]. On the one hand, the development of 5G standards and technologies does not only aim at providing effective communications to end-users. Whereas this was a primary driver in the previous generations of mobile networks, 5G considers vertical sectors as key adopters, and aspires to create a novel ecosystem where technical innovations and business models are also driven by vertical-specific use cases. On the other hand, UAVs are currently gaining prominence as key assets in different vertical markets, such as smart-farming, smart-cities, and public-safety.

In these contexts, UAVs have traditionally been considered as appropriate platforms to generate, process and/or transport relevant information (*e.g.*, video and other sensed data). This has been fostered with the recent advancements on the miniaturization of electronic devices, which allows UAVs on-board lightweight hardware platforms. These platforms open new and exciting opportunities to support a cost-effective and flexible provision of vertical applications. As an example, a smart-farming provider could use a number of UAVs to rapidly build an aerial network infrastructure over an extension of farmland. These UAVs could support the collection, generation, processing, and dissemination of relevant information to offer smart-farming operations, *e.g.*, surveying and evaluating the status of a crop field, or detecting objects of interest to perform closer inspection, or aid tasks such as precision spraying. In a different environment, an emergency service provider could rapidly deploy a fleet of UAVs to support voice and data communications to a team of firefighters, working on a fire extinction activity along a large forest area, where existing communication infrastructures may be insufficient or even unavailable.

Notwithstanding the availability of UAV devices to incorporate additional hardware platforms, this view still presents a fundamental challenge: the lack of flexibility exposed by existing UAV products. The UAVs are commonly equipped for specific purpose missions with a fixed set of software and hardware appliances, and base their operation on proprietary mechanisms and protocols. It is only when a programmable logic is embedded into the aerial vehicles that the flexibility of multi-mission UAVs can be freely expressed (see Chapter 4). On the one hand, a programmable logic might allow the provision and configuration of different functions and services over adaptable UAV units (*e.g.*, to act as an aerial relay for voice and data communications). On the other hand, such a logic could enable the flexible interconnection of UAVs to build programmable aerial networks, which could be rapidly deployed over delimited geographic areas. Based on these observations, the prior work presented in Chapter 4 explored the utilization of Network Functions Virtualization (NFV), a key technology in 5G networking, to provide such programmable logic. This work gave the first steps towards validating the practical feasibility of this approach, showcasing the deployment of a functional IP telephony service

over an NFV system based on UAVs.

Following the above mentioned research line, this chapter studies the practical feasibility of utilizing programmable UAVs to support heterogeneous vertical services. To this purpose, this chapter builds on the UAVs-based NFV system presented in Chapter 4, and on the mechanisms described in Chapter 3 related to the orchestration of telecommunications and vertical services within a multi-site NFV ecosystem. In this context, this chapter presents the design and development of an experimentation testbed at national scale, spanning three different (remote) sites, which was set up with the main goal of facilitating experimentation with vertical services, with UAVs allowing for the prototyping and validation of these services in a realistic multi-site environment.

5.2. Description of the Distributed NFV Testbed

The distributed NFV testbed was built taking advantage of the joint efforts of different Spanish universities, *i.e.*,Universidad Carlos III de Madrid (UC3M), Universidad Politécnica de Cataluña (UPC), and Universidad del País Vasco (UPV/EHU) together with the 5G Telefonica Open Network Innovation Centre (5TONIC) laboratory [63], based in Madrid. The following subsections describe the infrastructure and services at each experimentation site, as well as the mechanisms that support their interconnection and joint operation. It is important to note that the work of the author of this thesis, in relation to the creation of the distributed testbed, is limited to the context of the 5TONIC/UC3M site, and to the application of the mechanisms to integrate the two remaining sites (*i.e.*, UPC, and UPV/EHU). In any case, the description of the latter two sites is included below for the sake of completeness. An overview of the whole NFV ecosystem is depicted in Figure 5.1.

5.2.1. The 5TONIC/UC3M site

As a member of 5TONIC, UC3M (in particular, the author of this thesis) created an NFV experimentation infrastructure that could be flexibly interconnected and operate in coordination with other sites holding NFV capable equipment. This experimentation infrastructure was built from the research results obtained with the previous chapters of this thesis (see Chapter 3, Chapter 4), and was made available at the 5TONIC premises.

The development of the 5TONIC/UC3M site included a Management & Orchestration (MANO) system based on two open source and widely adopted software components: an Open Source MANO (OSM) [27] software stack, which provided the functionalities of an NFV Orchestrator (NFVO) and a VNF Manager (VNFM), and an OpenStack controller [32], acting as a Virtualized Infrastructure Manager (VIM). Both components ran in a single server computer, each in an independent virtual machine. As already described in Chapter 3, this approach facilitated the independent evolution of each component, as the software base of OSM and OpenStack evolved, as well as their vertical scaling in terms of allocated resources. Indeed, the most recent version of OSM (*i.e.*, OSM Release SEVEN) available at the time was installed for this testbed, maintaining previous installed versions for other use

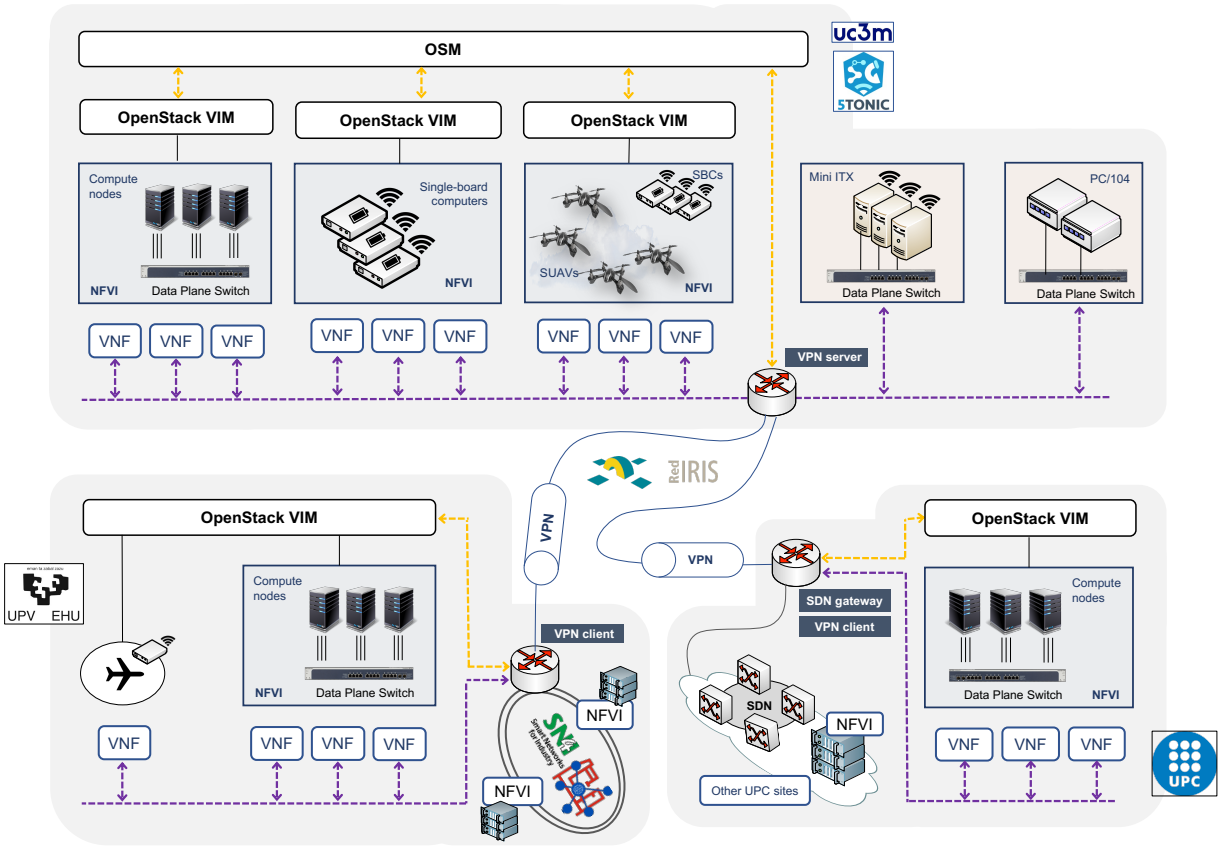


Figure 5.1. *Overview of the distributed NFV testbed.*

cases and/or projects. Thus, the MANO system controlled a set of compute, storage and network resources provided by a cloud of three interconnected server computers, each equipped with four Gbps Ethernet ports. The server computers account for a total of 24 vCentral Processing Unit (CPU)s, 96 GB of RAM, and 6 TB of storage.

In addition, this experimentation site involved, based on the design presented in Chapter 4, the creation of a resource-constrained NFV Infrastructure (NFVI). This infrastructure encompassed a set of ten battery-powered Single-Board Computers (SBCs), Raspberry Pi 3 Model B+, each with a 100 Mb/s Ethernet interface and several Wi-Fi adapters. Due to their characteristics in terms of size and weight, these SBCs represented a hardware platform that could be conveniently used in UAV and constrained device applications. The SBCs might be flexibly interconnected to form different multi-hop wireless network topologies, according to any specific experiment requirements. These SBCs were integrated into the NFV ecosystem following the approach specified in the prior work outlined in Chapter 4. This way, the experimentation resources provided by the SBCs were accessible to the OSM stack of 5TONIC via a separate OpenStack VIM, and VNFs could be automatically executed onto the SBCs using container virtualization technologies (LXC/LXD).

The 5TONIC/UC3M site integrated an additional portable UAV-specific NFV infrastructure, which evolved from the platform presented in Chapter 4 to support different experiments. This infrastruc-

ture consisted of four small-size UAVs (model Parrot Bebop 2 [111]). Each UAV on-boarded an SBC with two Wi-Fi interfaces (a Raspberry Pi 3 Model B+ with an additional Wi-Fi USB adapter). This infrastructure could be complemented as needed with the battery-powered SBCs that are available in the site, which could represent ground units for the purposes of experimentation (*e.g.*, UAVs perched on the ground or other terrestrial vehicles). All the SBCs involved in an experiment could be attached as compute nodes to a portable OpenStack VIM that ran on a laptop. This VIM offered an interface to the OSM stack to handle the deployment of VNFs over the whole infrastructure of SBCs. Due to regulatory reasons, tests with real UAV equipment at 5TONIC could only be carried out indoor, using specific locations that have been made available for this purpose.

Finally, the construction of 5TONIC/UC3M site also included two modular computer platforms conforming to the PC/104 embedded computer standard, each with one 8-port Gbps Ethernet switch, enabling the interconnection of local equipment and applications at an UAV unit, and two Gbps Ethernet ports, to support the exchange of information with external entities (*e.g.*, with a ground control station in a realistic scenario). These platforms were designed to support IP communications in a tactical surveillance UAV from the Spanish Ministry of Defense [100]. In addition, a testbed with ten mini-ITX computers with Linux Operating System (OS) served several purposes. For instance, to complement the resources of the cloud computing infrastructure offered by the SBCs, and support the execution of VNFs that require more powerful equipment. Or to provide the functionalities of wireless routers and/or access points, *e.g.*, with OpenFlow [121] support.

The physical disposition of components in the 5TONIC/UC3M site allowed the flexible and on-demand interconnection of the aforementioned equipment, according to the experimentation needs. This enabled experimentation with diverse and heterogeneous VNFs (*e.g.*, lightweight VNFs) and physical network functions (*e.g.*, mini-ITX or PC/104 computers), which could exchange data through the network infrastructure available at 5TONIC/UC3M site.

5.2.2. The UPV/EHU site

The infrastructure provided by UPV/EHU (illustrated in the lower left corner of Figure 5.1) comprises a cloud platform implemented with OpenStack. In addition, this infrastructure includes a limited computing device based on a Raspberry Pi 3 Model B+, which can be on-boarded into an UAV, and executes the system that supports the communications of the UAV with the infrastructure. All the technical details described by UPV/EHU in relation to this part of the distributed NFV testbed are available in [122].

5.2.3. The UPC site

With respect to the UPC infrastructure (represented in the lower right corner of Figure 5.1), it consists of three cloud platforms implemented with OpenStack, which provide a multi-site experimentation environment. One of these platforms was allocated to be part of the multi-site NFV testbed developed

within the context of this chapter, and all the technical aspects described by UPC in regards to this infrastructure are also available in [122]

5.2.4. *Inter-site communications and joint operation*

To integrate the three experimentation sites into a single NFV ecosystem, and support their joint operation, an overlay network was created to support inter-site communications. This was supported through the Virtual Private Network (VPN) service previously described in Chapter 3, hosted at 5TONIC. This service enabled the configuration of secure point-to-point virtual links between 5TONIC and every other site, each of them with a gateway function that ran a VPN client. Data transmitted over these virtual links were physically disseminated towards their proper destinations over the high-speed backbone network of RedIRIS [123]. The VPN server at 5TONIC behaved as a network router, relaying data traffic across the point-to-point virtual links, which guaranteed end-to-end network connectivity among the different sites.

The coordination of orchestration actions among the three sites was guaranteed with the utilization of a single NFV orchestrator, which was provided by the OSM stack installed at 5TONIC. The OSM software provided a northbound interface that could be used to support common NFV operations, including the on-boarding of VNF and service descriptors, and the commission and termination of multi/single-site network services composed by different VNFs. The latter was achieved through Management & Orchestration-specific communications, which took place between the OSM stack at 5TONIC and the VIMs available at the three experimentation sites (see Section 3.3.2 for more details on the communication scheme).

In the distributed NFV testbed, the VPN-based overlay network played a fundamental role to support orchestration actions, enabling the distribution of the following types of information: *(i)* control communications between the OSM stack and the remote VIMs, to support the management of the compute, storage and network resources available at each site; *(ii)* management communications to support the configuration of Virtualized Network Function (VNF)s after their deployment; and *(iii)* inter-site data communications, to support the exchange of information among VNFs running at different sites.

5.3. Experimental Validation

This section aims at validating, through the multi-site NFV experimentation testbed created, the utilization of the NFV system based on UAVs (defined within the scope of this thesis) in the context of different vertical environments. The primary objective is to verify the ability of the setup for facilitating experimentation in the context of vertical services, with UAVs allowing for the prototyping, testing and validation of these services in a realistic multi-site environment. In particular, the following lines present the realization of a specific use case of a smart-farming vertical, describing the main aspects

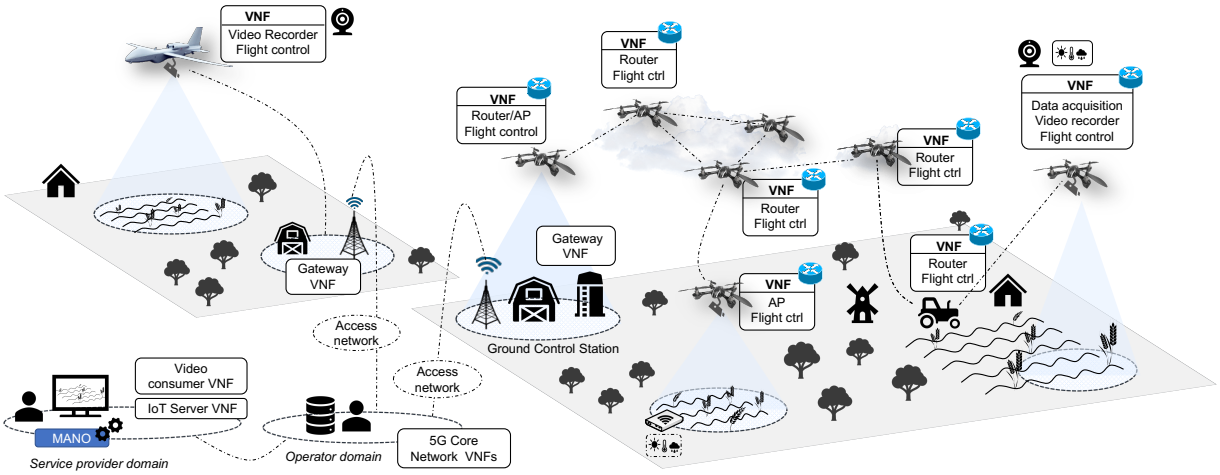


Figure 5.2. *Outline of the vertical use case.*

regarding the experimental setup, and summarizing the results obtained during the functional validation.

5.3.1. Definition of the vertical use case

UAVs have lately gained attention as enabling platforms to support the concept of smart-farming. An UAV can carry different payloads, such as high-resolution daylight video cameras, thermal imaging cameras, Global Positioning Systems (GPS), and a myriad of other low-cost sensor and electronic devices. These include wireless transceivers (*e.g.*, Wi-Fi, line-of-sight radio, or 3G/4G), to support network communications towards ground units and other aerial vehicles. This way, this new type of devices emerged as a suitable platform to facilitate observation and remote sensing operations over crop fields.

On the one hand, the design space of smart-farming applications has considered the utilization of standalone UAV units. Based on information that can be acquired by an UAV, by means of the sensors and the electronic devices that it can carry as payload, aerial vehicles may serve multiple purposes including, among others, surveying of the agricultural field [124], the evaluation of water status in a crop field [50], and the detection of objects of interest in agricultural environments through computer vision, *e.g.*, to perform closer inspection or precision spraying [125]. But other research studies also contemplate the deployment of UAV swarms to enhance the situational awareness on the farmland via cooperative and/or simultaneous operation, *e.g.*, to support the collaborative generation of relevant images [126], or to increase the precision of monitoring operations over delimited areas [51].

To verify the capacity of UAVs-based NFV system to support experimentation activities with smart-farming services, the reference use case depicted in Figure 5.2 was considered. In this use case, a smart-farming service provider deploys a number of UAVs over an extension of farmland. These UAVs are configured to provide functions specific to smart-farming, such as collecting data from sensors

deployed over the crop field, and recording high-resolution daylight/thermal video/images, relaying all this information towards a remote equipment owned by the service provider. These functions are implemented in software and deployed on these UAVs on demand, as VNFs, using a MANO system under the control of the service provider. The collected information may serve different purposes, like the early identification of diseases, the estimation of the production volumes, *etc.*

In this reference use case, a *Flight Control VNF* could be instantiated at each aerial vehicle, being configured after its deployment with information on fixed trajectories in the form of way-points, to be autonomously followed by the vehicle. Real-time control of an UAV by a human operator could also be supported by that *Flight Control VNF*, for instance to enable a more detailed examination of a given zone of the farmland due to the detection of a potential disease. The information produced by the electronic devices on-boarded on the UAVs (*e.g.*, live video/images and other sensed data), as well as information collected from sensors on the ground, would be delivered to a gateway function in a ground control station (a facility close to the farmland in charge of the maintenance of the UAVs). In addition the reference use case considers that UAVs are flying over a remote area, with limited access to communication infrastructures. Therefore, the collected information is disseminated towards the ground control station through a multi-hop wireless network conformed by the UAVs themselves (flying or perched on land, for instance in specific-purpose ground structures). To this purpose, UAVs execute the functions of wireless access points and/or routers as VNFs. Other devices in the farmland (*e.g.*, harvesters, tractors, sprayers, *etc.*) might also be opportunistically used to provide similar functions. The information received by the gateway function would be relayed towards the service provider operating the deployment, via a fixed or a cellular (3GPP or non-3GPP) access network. In a realistic scenario, this would be supported by a telecommunication operator, who could offer 5G core network services under the control of a MANO platform. To facilitate the realization of the use case and provide an experimental setup, it is consider a single MANO platform, as indicated in Section 5.2.4.

In the reference scenario, a larger UAV equipped with a thermal imaging camera flies at a higher altitude, recording images that may serve to inspect and monitor the crop field and detect potential diseases. These images would be transmitted to the service provider through a second ground control station, which maintains a line-of-sight radio link with the UAV.

At the service provider location, relevant information (*e.g.*, sensed data, video, images, *etc.*) would be processed and stored in specific-purpose servers, which would also be provisioned as VNFs. This information could also be inspected in real-time by a human operator.

5.3.2. Experimental setup

To evaluate the feasibility of utilizing virtualization technologies and resource-constrained platforms to support a smart-farming use case, the distributed NFV testbed presented above (see Section 5.2) was used to build the experimental setup shown in Figure 5.3, which is aligned with the use case presented in the previous section.

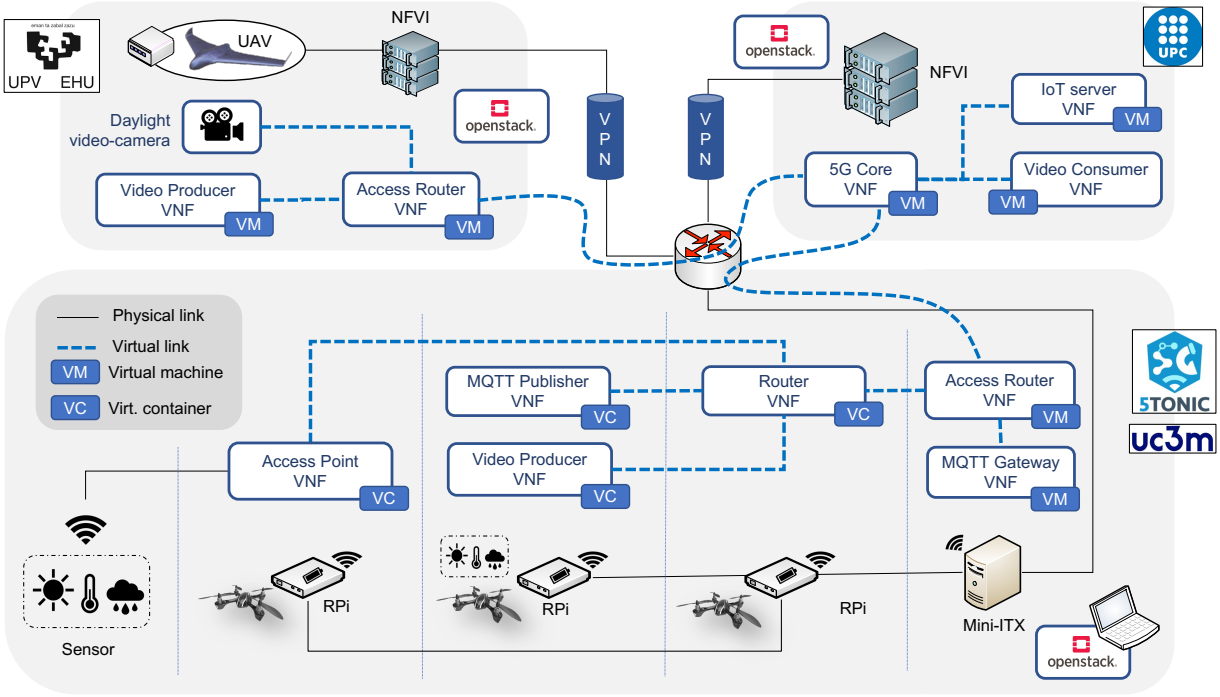


Figure 5.3. *Definition of the smart-farming NFV service and experimental setup.*

The 5TONIC/UC3M site contributed with three UAVs Parrot Bebop 2, each on-boarding a compute node (*i.e.*, a Raspberry Pi 3 model B+). A Wi-Fi access point, deployed at one of these compute nodes as a VNF (labelled in Figure 5.3 as *Access Point VNF*), provided network access connectivity to a ground equipment with a sensor that measures humidity, pressure, and temperature. The information read by this sensor was sent to a gateway function deployed as a VNF on a ground compute node (*i.e.*, a mini-ITX computer). The gateway function (named as *MQTT Gateway VNF*) retrieved data from the sensor using the Message Queuing Telemetry Transport (MQTT) protocol [127], being the information delivered through the Wi-Fi access point VNF and a router VNF (deployed on another UAV, and referred to as *Router VNF*). To emulate the generation of information by electronic devices on-boarded onto an UAV, the experimental setup included two additional VNFs: an *MQTT Publisher VNF*, which generated random temperature values (integers between 30 and 35 °C) and transmitted them to the gateway VNF; and a *Video Producer VNF*, which emulated the generation of a real-time video as if it were generated by a video camera at the UAV. This VNF was implemented using an open source traffic scheduler, Trafic [128]. It is important to highlight that this experimental setup was created within a laboratory environment with indoor controlled flight conditions, with a main focus on the network and vertical-specific functions that are needed to support the collection, dissemination, processing, and storage of relevant data in a smart-farming use case. The implementation of specific functions for autonomous flight control, which might be needed in realistic scenarios, is out of the scope of this chapter.

In the experimental setup, a second video source was provided at the UPV/EHU site, where a

reconfigurable node was instructed to transmit a real-time video produced by a high-resolution daylight video camera. This served to represent a video recorded by an UAV flying at a higher altitude. The video produced by this camera was disseminated using the Real-Time Transport Protocol (RTP) [129]. Alternatively, the video generated by this UAV might be emulated by a *Video Producer VNF*, which was also available at the UPV/EHU site. This enabled test procedures involving the transmission of real-time video from the UAV at different qualities. In this sense, the experiments presented below involved the high-resolution daylight video camera.

Real-time video flows, originated at the 5TONIC/UC3M and the UPV/EHU sites, were delivered to a *Video Consumer VNF* at the UPC site. This VNF, which was built using Trafic and the VLC media player, handled the received video information to obtain performance metrics for validation purposes. In addition, the UPC site held an *Internet-of-Things (IoT) server VNF*, which uses the MQTT protocol to retrieve all the data collected by the *MQTT Gateway VNF*, making these data available to authorized parties through a web interface. The *IoT server VNF* was developed using an open source IoT platform, MainFlux [130].

Finally, the experimental setup included the emulation of the information that might be collected and/or relayed by UAVs, which were sent towards the smart-farming service provider through a non-3GPP cellular access network. To this purpose an *Access Router VNF* is deployed at the 5TONIC/UC3M site and the UPV/EHU site. This VNF supported the user-plane protocol stack defined by 3GPP for untrusted non-3GPP accesses [116], which is based on the GRE [118] and IPsec [119] standard Internet protocols. The *Access Router VNFs* encapsulated the information transmitted from each site over a GRE/IPsec tunnel, sending it towards a *5G Core VNF* running at UPC. The *5G Core VNF* was developed to support the same protocol stack as the access routers. Hence, it decapsulated the information received from the GRE/IPsec tunnel, forwarding it to the *IoT server VNF* or the *Video Consumer VNF* as appropriate.

5.3.3. Functional validation

The MANO system at the 5TONIC/UC3M site was used to automatically deploy and configure the vertical service shown in Figure 5.3 (all the NFV descriptors and images were made available as open source software [131]). To support the automated configuration of all the VNFs, specific Ansible playbooks were developed since their support by the OSM software stack due to the contribution presented in Chapter 3 (for more detail, review Section 3.3.4). Configuration aspects included, among other things, the definition of appropriate IP addresses and network routes in the case of access points and routing functions, and the activation of the services that might be needed to provision functional VNFs (*e.g.*, starting the MainFlux application in the *IoT server VNF*).

Once the vertical service were successfully deployed and configured, all the sensed data (either real or emulated) started to be disseminated towards the *IoT Server VNF*. Analogously, the *Video Consumer VNFs* started receiving the video from the 5TONIC/UC3M and the UPV/EHU sites. The UAV executing the *Router VNF* was pre-configured to fly (indoor) hovering over a static position for the

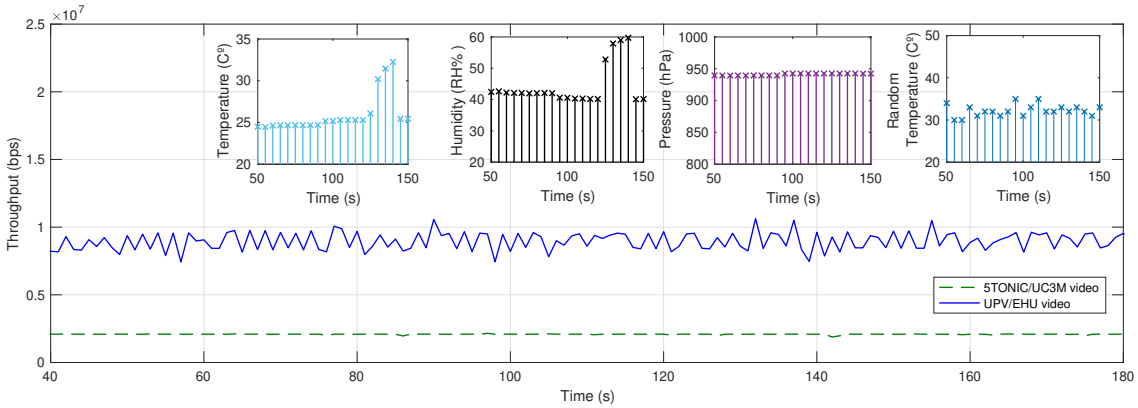


Figure 5.4. *Functional behavior of the smart-farming service.*

whole duration of the experiment. This served to verify that the UAVs could fly while carrying an SBC as payload.

Figure 5.4 presents the average throughput of both videos, measured at the *Video Consumer VNF*, during a time interval of 180 seconds. They were received uninterruptedly, with an average rate of approximately 2 Mb/s and 9 Mb/s (their corresponding average sending rates). Additionally, the small frames in the top part of the figure represent a 100 seconds interval of temperature, humidity and pressure readings from the real sensor that was available in the experimental setup, as well as random temperature readings provided by the *MQTT Publisher VNF* in the same time interval. Readings were provided by the real sensor at a low rate, with a temperature, humidity and pressure value transmitted towards the *IoT Server VNF* every 5 seconds (random temperature values were also provided at the same pace). No sensed data was lost during the experiment. These results validated the functional behavior of the multi-site smart-farming service depicted in Figure 5.3, and suggested the potential of using virtualization technologies and UAV platforms to support vertical-specific applications.

As previously commented, a very relevant advantage of programmable UAV platforms is their ability to support the rapid deployment of different vertical services. In this respect, the service creation time cycle was recognized as a paramount performance parameter in the context of 5G networking. In particular, according to the 5G Infrastructure Public Private Partnership (5G-PPP) view, it must be reduced to an average of 90 minutes [3].

To verify the service creation times that could be achieved by this NFV setup for moderately-complex multi-site vertical services, 30 consecutive deployments of the vertical service defined in Figure 5.3 were performed. The deployment delays for this multi-site service are represented in the boxplot of Figure 5.4. The figure also presents the time delays that are needed to deploy the vertical service at each site (5TONIC/UC3M, UPV/EHU, and UPC). To estimate these time delays, the functional validation included the creation of an NFV vertical service descriptor per site, including only the VNFs that are to be deployed at that site (*i.e.*, 6 VNFs at 5TONIC/UC3M, 2 VNFs at UPV/EHU, and 3 VNFs at UPC). Each one of the three vertical service segments was deployed 30 times at its corresponding site, leading to the delay values presented in the boxplot chart.

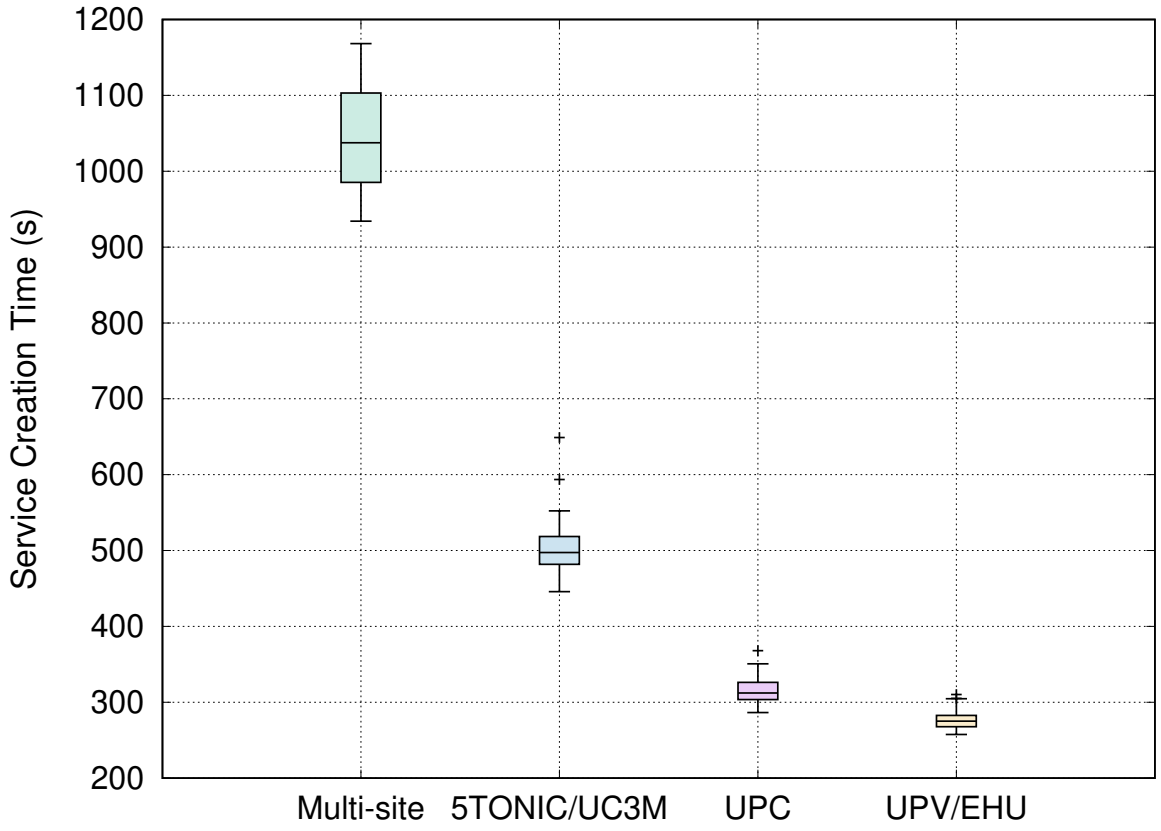


Figure 5.5. *Deployment times of the smart-farming service.*

The average service deployment delay for the multi-site vertical service was 1,052.2 s (less than 18 minutes). This represents an adequate performance figure in line with the previously mentioned 5G-PPP view. On the one hand, it enables the deployment of a moderately complex vertical service over several cloud and portable resource-constrained NFV infrastructures. On the other hand, it leaves sufficient time to carry out the flight procedures that may be needed in real scenarios, to position UAV units at their corresponding locations. The average deployment times for the vertical service segments at 5TONIC/UC3M, UPV/EHU, and UPC were 505.3 s, 277.7 s, and 315.8 s, respectively. From these values, it can be observed that service deployment time increases with the number of VNFs, as it was expected. In addition, the average time required to deploy the multi-site vertical service (1,052.2 s) is lower than the aggregation of the deployment delays of the three vertical service segments (accounting for a total of 1,098.8 s). This suggests that OSM provides a certain degree of concurrency in the instantiation of NFV services.

At this stage, it is important to note that the deployment time for the multi-site vertical service under consideration includes two main components: *(i)* the time needed to instantiate and interconnect the required virtual machines and virtualization containers; and *(ii)* the delay incurred in performing their configuration through Ansible playbooks to provide functional VNFs. To gain a better understanding on the impact of each of these components on the service deployment times, an additional

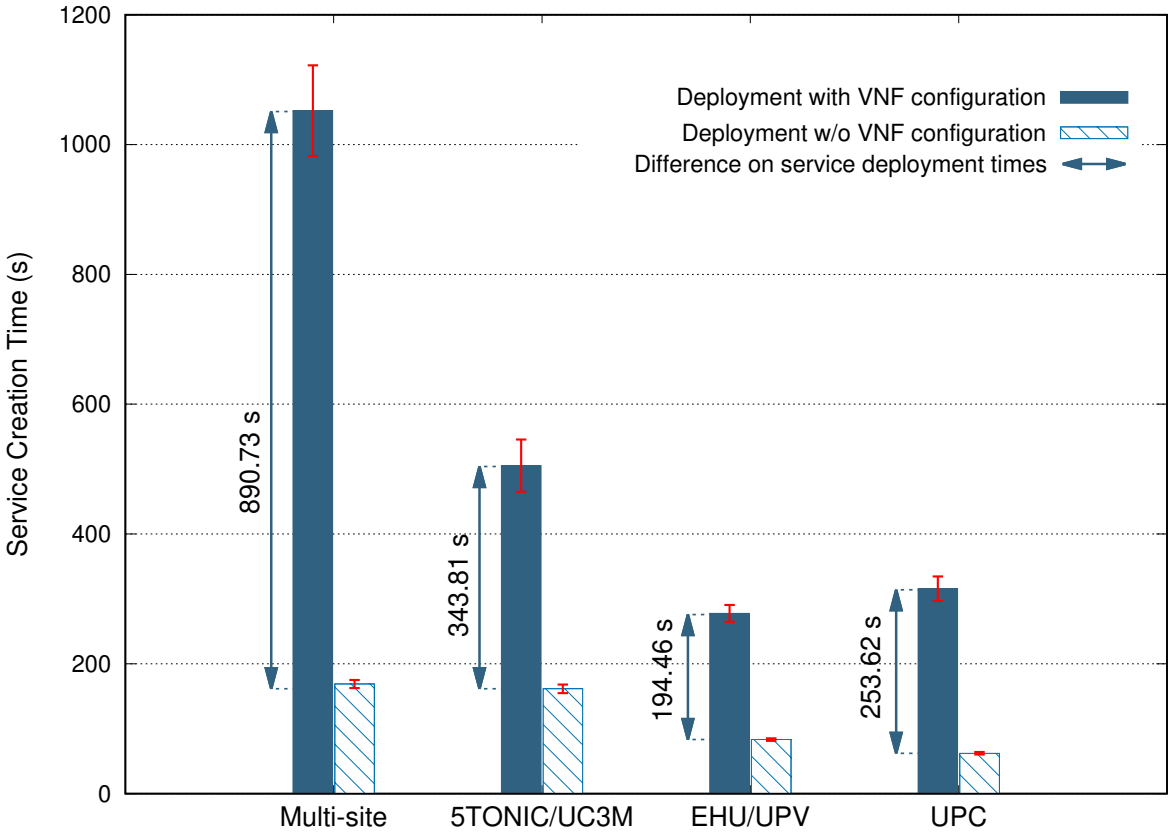


Figure 5.6. *Deployment delays with and without VNF configuration.*

experiment was performed. New versions of the vertical services of Figure 5.5, which do not execute VNF configuration procedures (*i.e.*, the deployment of these services only involves the instantiation and interconnection of their corresponding virtual machines and virtualization containers), were instantiated. Figure 5.6 shows the average deployment times of such services, comparing them with the average delays that were previously obtained when VNF configuration procedures were executed. From the figure, it can be observed that the deployment time of the services significantly increases when their corresponding VNFs are subject to configuration.

The reason for this resides in the mechanism of common use in OSM to carry out the configuration of VNFs. OSM supports a limited form of Juju charms [36], referred to as VNF configuration charms or proxy charms. Once the OSM software stack requests the VIM to instantiate the virtual machines and virtualization containers needed by the VNFs, and while these instantiations are in process, a Juju agent deploys a proxy charm for each of the VNFs that require configuration. A proxy charm is basically a collection of scripts and software that are specified as part of the VNF descriptor. In particular, the proxy charms implemented for the smart-farming use case support the remote access to the VNFs via Secure Shell (SSH) protocol, and perform their configuration through Ansible playbooks (these can also be specified as part of the VNF descriptors). The Juju agent installs each proxy charm on a Linux container, and provides an interface to the OSM software stack to handle the execution of the

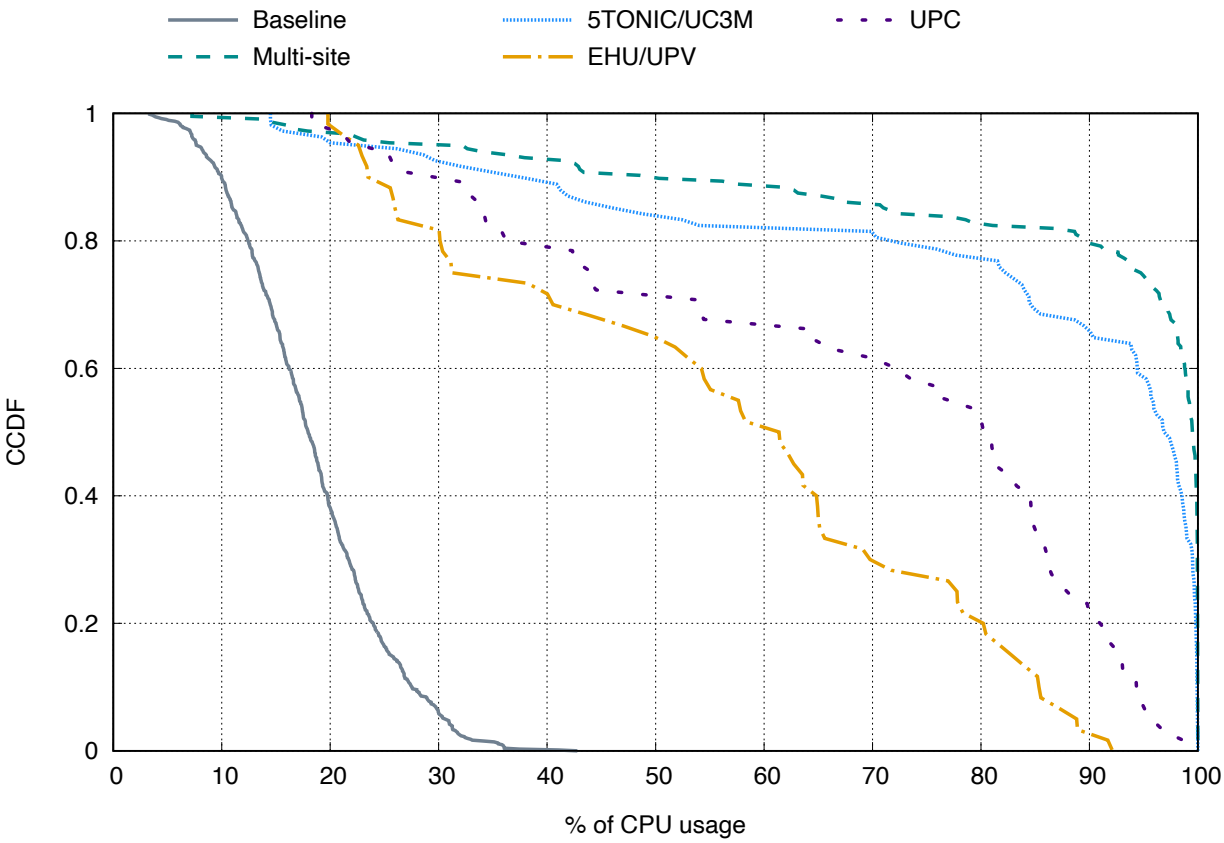


Figure 5.7. *CCDF of CPU utilization during service deployments.*

scripts bundled within the charm. This allows the automated configuration of a VNF. The OSM software stack monitors the outcome of the VNF configuration tasks, through status information reported by the Juju agent.

This is an effective mechanism to support the configuration of VNFs, which also provides a high degree of flexibility to VNF developers in defining configuration actions. Still, the experimental results indicate that the complexity of the configuration processes, required to provision the Linux containers and to synchronize the execution of the configuration primitives, results in a significant increase of service deployment times and processing load. The latter is illustrated in Figure 5.7, which shows the Complementary Cumulative Distribution Function (CCDF) of the average CPU utilization during one deployment of the multi-site vertical service, measured in 5 second intervals at the virtual machine that hosts the OSM stack. From the average CPU utilization samples, it can be observed that the configuration process of the VNFs imposed a significant load to the OSM virtual machine, with 79.63% of the samples taken showing an average CPU utilization higher than 90%. During 56.94% of the deployment time, the average CPU utilization exceeded 99%. This suggests that the efficiency of VNF configuration based on proxy charms still has room for improvement.

To observe the dependency between the CPU utilization and the number of VNFs of the vertical service, the figure also represents the CCDF of the average CPU utilization for a single deployment

Vertical Service	**Nº of VNFs**	**Average CPU utilization**	**CCDF (CPU 99%)**	**CCDF (CPU 90%)**
Multi-site	11	89.46	0.5694	0.7963
5TONIC/UC3M	6	84.18	0.3333	0.6574
UPC	3	69.19	0	0.20
EHU/UPV	2	57.12	0	0.0333

Table 5.1. *CPU utilization during a single deployment of each vertical service.*

of each vertical service segment, measured at 5 second intervals. It also shows a complementary cumulative distribution for a reference scenario, in which no NFV deployments were conducted by the OSM stack during a period of 1 hour. Table 5.1 summarizes some of the relevant values collected during these four deployments.

As it was expected, the average CPU utilization declines with the number of deployed VNFs. Moreover, the distribution of CPU utilization exhibits a trend towards the baseline scenario as the number of deployed VNFs decreases. In the case of the 5TONIC/UC3M deployment, there was still a noticeable degree of overload, with the average CPU utilization exceeding 99% for approximately one third of the deployment time. This behavior is related with the VNF configuration mechanism detailed before, which increases the requested amount of CPU utilization with the number of VNFs to be configured. In the case of UPC and UPV/EHU, the average CPU utilization was lower, with reduced periods over 90% (20% of the deployment time for UPC, and 3.33% for UPV/EHU). Even in that cases, with lower CPU utilization values, the execution of VNF configuration primitives imposes a substantial delay to service deployment, as observed in Figure 5.6.

5.4. Conclusions

The work presented throughout this chapter is related to explore the utilization of the UAVs-based NFV system presented throughout this thesis to support 5G vertical services. As a result of this study, this chapter has presented the following main contributions:

- Design and development of multi-site NFV experimentation testbed spanning three different and remote sites: 5TONIC/UC3M, UPC, UPV/EHU. This testbed was based on the research findings of the previous chapters of this thesis, and supported the execution of the validation tasks.

- Definition of a use case related to the smart-farming vertical, to validate the utilization of the NFV system based on UAVs in the context of vertical environments.

- Initial approximation to evaluate, by means of the experiments performed with the smart-farming use case, the service deployment times that can be achieved in UAV-based vertical use cases, using existing open source NFV technologies.

The research results obtained in the context of this chapter suggest the potential of the NFV system based on UAVs to support the flexible execution of moderately complex vertical services. However, these experiments also indicated that the execution of VNF configuration primitives may impose a significant CPU workload to the MANO software stack. This could be a symptom of problematic situations in production NFV environments, hindering the capacity of theMANO platform to perform concurrent operations, such as managing the lifecycle of multiple moderately complex vertical services.

Under this perspective, the following chapter considers these aspects, and addresses the design and implementation of lightweight mechanisms for Ansible-based VNF configuration, aiming at: *(i)* distributing the configuration burden among sites where VNFs are actually deployed; *(ii)* reducing the execution delays of configuration primitives; and *(iii)* decreasing the effective computational load of the MANO stack. The latter will enable to better support certain UAV scenarios, where the deployment of services must be satisfied with very stringent time constraints (*e.g.*, in emergency use cases related to a public-safety vertical).

CHAPTER 6

Using Aerial and Vehicular NFV Infrastructures to Agilely Create Vertical Services

5G communications have become an enabler for the creation of new and more complex networking scenarios, bringing together different vertical ecosystems. As mentioned in previous chapters, such behavior has been fostered by virtualization and softwarization technologies (*e.g.*, Network Functions Virtualization (NFV), and/or Software Defined Networking (SDN)), where the orchestration and virtualization capabilities allow the possibility of dynamically supplying network resources according to its needs. Nevertheless, the integration and performance of heterogeneous network environments, each one supported by different providers, and with very particular characteristics and requirements (*e.g.*, resource-constrained infrastructures), in a single framework is not straightforward.

On this basis, this chapter follows the research line of the previous one, considering other vertical sectors, and exploring the potential of the NFV system based on Unmanned Aerial Vehicles (UAVs) to interoperate with other NFV infrastructures, and support the deployment of telecommunications and/or vertical services in resource-constrained situations. In particular, it considers an existing research development conducted by the Instituto de Telecomunicações of Aveiro (Portugal), which employs an NFV infrastructure based on a fleet of vehicles. This work is proposed as a collaboration with this institute with the aim of studying the combined utilization of UAV and vehicular NFV infrastructures to provide vertical services in environments with significant resource limitations. As a result of this collaboration, this chapter presents the development of a common framework that relies on the previous research findings of this thesis related to the UAVs-based NFV system, and to the mechanisms enabling the operations of a multi-site NFV ecosystem.

For this, the main conceptual aspects in respect to the definition of the framework, in which the computational resources are provided by mobile devices, are addressed in Section 6.2. Section 6.3 delves into the applicability of the framework proposed by means of defining a significantly complex scenario, with its main architectural blocks detailed. In this context, a use case involving the public-safety vertical will be used as an illustrative example to showcase the practicality and potential benefits of exporting the UAVs-based NFV system to different environments. This chapter also includes the technical implementation details of the framework proposed, allowing to analyze and discuss the delays on the network services deployment process. In this vertical, the rapid configuration of Virtualized Network Functions (VNFs) is required. However, following the traditional model of service provisioning, where the orchestrator is usually centralized, the configuration of VNFs may incur relatively high delays. Section 6.4 depicts the evaluation results related to this analysis, verifying that deployment times are relatively high, and proposes and validates a solution to solve this problem. Finally, Section 6.5 presents the main conclusions of this chapter.

6.1. Introduction

Over the last few years, the transition to the new generation of mobile communications (*i.e.*, 5^{th} Generation of Mobile Networks, or 5G) has been gradually taking pace. To reach this current stage, the research community has carried out an arduous exercise to first define the requirements to be fulfilled by this new generation in order to address the direction taken by society's use of information technologies, and then, to establish the basis in the form of standards to be followed when developing the new generation technologies. In both cases, the research activity developed under the scope of projects funded by the European Union (EU), such as those corresponding to the different phases of the 5G Infrastructure Public Private Partnership (5G-PPP) initiative, or programmes such as the EU programme horizon 2020, have played a very significant role [132].

One of the most relevant challenges studied by these research activities is focused on realizing a paradigm shift concerning the model followed by the previous generations of mobile networks for the provision of communication services [2, 8]. Thus, 5G will contribute to a global digital transformation, involving diverse vertical sectors such as automotive, smart cities, healthcare or public-safety among others, to support the creation of an innovative ecosystem capable of accommodating advanced and modern developments in both technical and business domains. This novel ecosystem will bring a more comprehensive portfolio of services and applications with a resulting multiplicity of requirements beyond the current voice and mobile broadband, encompassing the massive connection of machine-type devices, high reliability, ultra-low latency and an enhanced mobile broadband with higher bandwidth [4, 5, 133]. Nonetheless, the full realization of this vision is challenging due to, among other reasons, the lack of flexibility to integrate heterogeneous network infrastructures with limited resources. This hampers the cost-effective provision of both vertical and telecommunication services in resource-constrained situations.

From this perspective, this chapter explores the utilization of the Network Functions Virtualiza-

tion (NFV) system based on Unmanned Aerial Vehicles (UAVs) to support the deployment of telecommunications and/or vertical services in resource-constrained situations. To this purpose, it analyzes the practicality and potential benefits of exporting the UAVs-based NFV system to different environments. In particular, it considers the automotive arena. This is motivated due to the existing research development, carried out by the Instituto de Telecomunicações of Aveiro (Portugal), which employs an NFV infrastructure based on a fleet of vehicles. This institute was a partner in the 5GINFIRE project [64], and based on the work carried out within the scope of this project, it studied the possibility of offering 5G services based on NFV over vehicular networks, applying the concepts previously presented in this thesis forUAVs, but to the vehicular environment [134–136]. This led to the collaboration with the institute in order to analyze the combined utilization of UAV and vehicular NFV infrastructures to provide vertical services in particular situations with resource constraints.

Taking into account the above considerations, this chapter presents the definition and implementation of a comprehensive framework capable of integrating dynamically heterogeneous NFV infrastructures distributed across different geographical locations in order to support the deployment of elaborated vertical services. Specifically, it analyzes how to carry out the integration between three NFV infrastructures with clearly differentiated capacities. In the first place, the framework envisages the integration of an infrastructure with high computing resources that allows the development of Virtualized Network Functions (VNFs) that can be spanned within a core domain of a 5G network. Then, the second infrastructure contemplated in this chapter incorporates into the framework the ability of destining on-demand the deployment of services in a flexible and automated manner wherever it is required, leveraging for this the inherent mobility capacity of UAV devices. The final infrastructure covered by the framework is an automotive environment capable of deploying opportunistically functionalities on real vehicles connected both with Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, so that end users in those vehicles may have enhanced access to the possible services to be deployed. It is worth mentioning that this work contemplates the integration of each and every one of the mentioned infrastructures under the same NFV Management & Orchestration (MANO) stack, which places the different components that make the stack up in a distributed way, alongside each infrastructure.

By means of this innovative joint integration, services and applications for specific verticals can be supplied. In this sense, and with the objective of corroborating the practicality of the proposed framework, the work of this chapter includes the definition of a scenario pertaining to the public-safety vertical that aims to monitor the state of a road given that possible adverse situations are foreseen. Finally, the chapter also includes the development of a novel solution of a configuration function based on the publish–subscribe model that can be incorporated into the MANO stack. Thus, addressing the agile configuration of the functionalities to be deployed in case of an emergency situation arising under the context of the public-safety vertical.

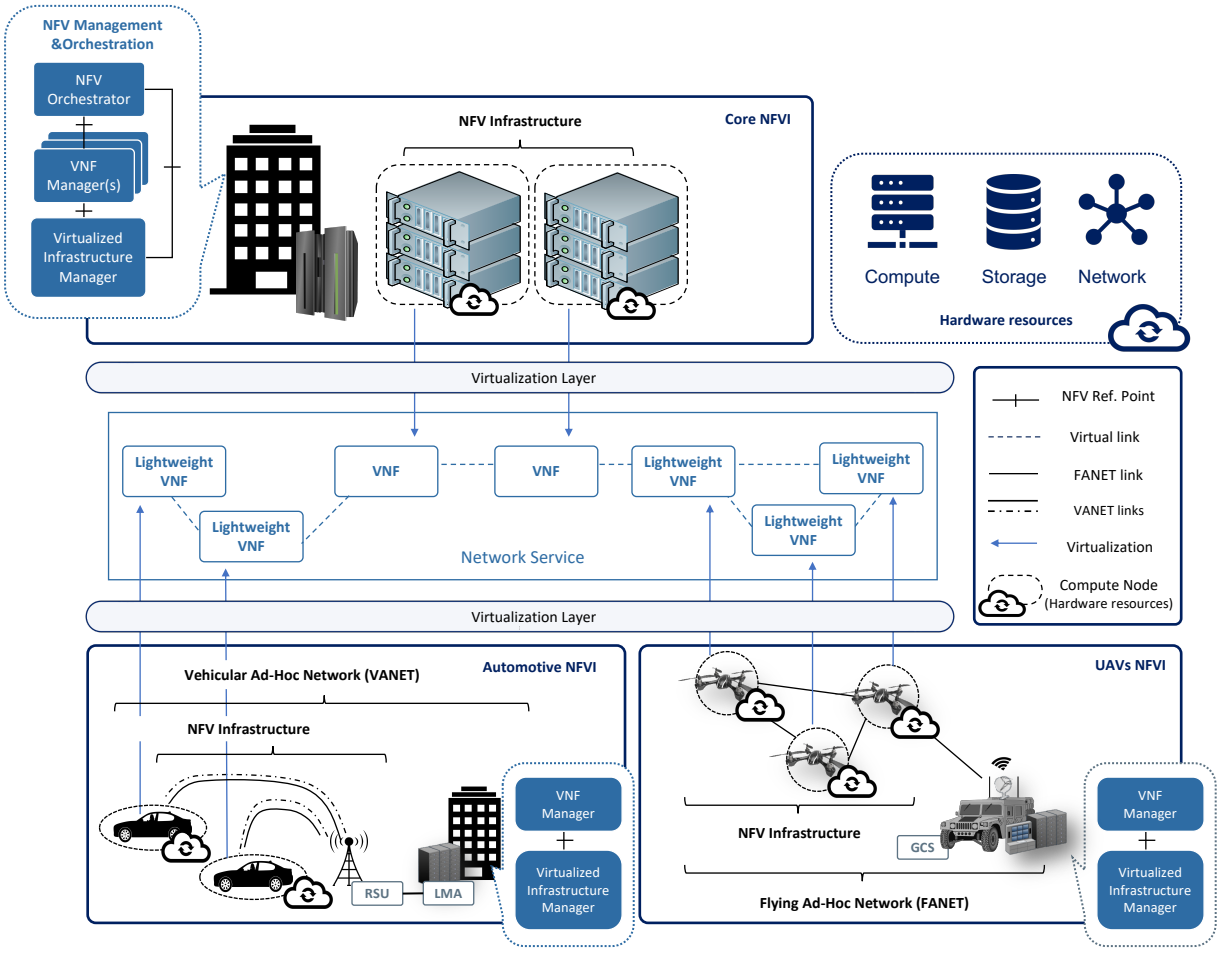


Figure 6.1. *Overall framework architecture.*

6.2. Description of the Aerial and Vehicular NFV framework

Inspired by the prior work independently carried out in both the UAV and automotive environments in order to support the flexible and automated deployment of network services through the use of the NFV technology, the work presented in this chapter defines an overall framework, aligned with the design principles of the NFV architectural framework published by European Telecommunications Standards Institute (ETSI) [26], to enable the creation of a more complete and distributed NFV ecosystem, supporting the flexible incorporation of diverse NFV infrastructures from distinct service providers, distributed across different geographical locations. Thus, being able to host complex communication services and applications. In this context, Figure 6.1 graphically shows the design bases of the entire framework.

The framework considers three different NFV Infrastructures (NFVIs), where the needed resources are available to support the deployment of VNFs that, through their interactivity, will result in the provisioning of different network services. First, the upper section of Figure 6.1 represents the so-called Core NFVI, with high availability of resources in terms of computing, network and storage to accom-

modate services corresponding to the core network domain. Alongside with this NFVI, the framework places part of the NFV MANO stack in charge of orchestrating and managing the deployment of the services that will be hosted within the whole ecosystem. On this basis, the Virtualized Infrastructure Manager (VIM) block within the MANO stack addresses the management and coordination of both hardware and virtual resources of this first NFVI. In conjunction with the VIM, the NFV Orchestrator (NFVO) and the VNF Manager (VNFM) are also located in the mentioned MANO stack. Such VNFM is in charge of supporting both the configuration of the VNFs hosted by the Core NFVI, and their lifecycle management, to provide to each virtualization unit with the expected functionality within a network service. For its part, the NFVO encompasses the lifecycle management of the network services, specifying how VNFs are connected to one another to form a network service, and triggering their deployment or depletion when it is required.

Up to this point, it has been simply described the part of the framework corresponding to a regular NFV system. Below, and in more detail, it is described how such system can be extended with two additional NFV infrastructures in which computation capabilities are provided by mobile devices, in particular, using UAVs and vehicles.

6.2.1. The UAVs NFV infrastructure

The lower right corner of Figure 6.1 reflects the component of the framework referred to as UAVs NFVI, following the conceptual design presented in Chapter 4 in which every UAV device comprises a computational unit that offers its hardware resources in terms of computation, storage, and networking, with the aim of enabling the execution of VNFs. As mentioned in previous chapters of this thesis, these resources are mainly limited due to the compact size of the UAVs, which also implies that they cannot carry any complementary hardware platform to greatly increase their computational capabilities to not compromise the flight operations. Due to this limitation on the available resources, the softwarization units (referred to as lightweight VNFs in the figure) have to implement their functionality in such a way that their execution does not involve a significant cost in terms of computing. Thus, endowing UAVs with the ability of adopting the virtualization paradigm introduced by the NFV technology, allows to provide an alternative, on-demand communication infrastructure wherever it is needed, either in areas where telecommunication infrastructure is not provided, or insufficient (*e.g.*, rural areas or areas severely damaged because of an emergency). This alternative also provides a high degree of flexibility when serving a wide range of functionalities, and their adaptation to the requirements imposed by particular interests of each occasion.

In this context, the VIM component, located in the Ground Control Station (GCS), coordinates the integration of different UAV units within the computing platform with the aim of comprising the UAVs NFVI as shown in Figure 6.1. In addition, this block is in charge of coordinating the available hardware resources and allocating them to the virtual resources that will fulfil the computing, storage and networking requirements of each lightweight VNF deployed into this infrastructure of the framework. The integration of this element within the MANO stack located in the Core NFVI enables the framework to orchestrate multi-site services in which both NFVIs will accommodate different VNFs

or lightweight VNFs capable of interoperating among themselves, with the added advantage that the UAV-based infrastructure can be positioned wherever required (or desired). Furthermore, the framework considers the possibility of including a VNFM element located in the GCS next to the VIM, so that, in case that the interoperation with the MANO stack of the Core NFVI is interrupted, the lifecycle of VNFs executed by the UAVs NFVI can continue to be managed. Finally, the implementation of the Flying Ad-hoc Network (FANET) represents a critical element to ensure the proper operation of the UAVs NFVI. The purpose of this component is threefold:

1) To enable the communications between the UAVs and the VIM to allow the latter to coordinate the operations regarding the NFVI (*i.e.*, manage the hardware and virtual resources).

2) To provide the underlying substrate on which the virtual networks will be created by the VIM in order to interconnect the VNFs and thus support the communications that will determine the functionality of the network service.

3) To allow the automated configuration of the VNFs, supporting the communications from the VNFM with the virtualization units once these have been provisioned by the VIM.

6.2.2. The Automotive NFV infrastructure

Figure 6.1 also depicts, in the lower left part of it, the last component of the overall framework, referred to as Automotive NFVI. This part of the framework was designed by the Instituto de Telecomunicações of Aveiro, and in this case, this infrastructure exploits the NFV technology to deploy virtualized applications close to the occupants of the vehicles (*i.e.*, inside the vehicles). To this purpose, this infrastructure is based on a Vehicular Ad-hoc Network (VANET), which not only provides connectivity to the vehicles, but also to the virtualized functions that are deployed in those vehicles. All the aspects related to the design of this infrastructure, including the main elements of the VANET on which it is based, were described in detail by the Instituto de Telecomunicações of Aveiro, and are available in [137].

6.3. Use Case Description

In the following section, the application of the proposed framework to a particular use case is defined, whose goal is to highlight potential scenarios for its implementation, as well as emphasising the significant benefits of such implementation. It firstly describes the use case, which is related to a public-safety vertical, and then it includes the details to be considered for the further implementation of the network service.

6.3.1. Initial vertical service deployment

With the beginning of the new era of mobile networks (or 5G, as it is commonly referred), an immense number of connected devices is envisioned to play an important role in the provision of application services to end-users. In particular, the automotive arena is considered as one of the most clearest examples in which the challenges specified for this new generation of communications (*e.g.*, reduced latency, high bandwidth or reliability, low energy consumption, *etc.*) need to be accomplished. Accordingly, the industry and the research community have presented how the technology improvements could be used to enable a secure, connected and automated driving [138].

In this context, the proposed use case considers a common situation where dense road traffic conditions can be expected in advance, *e.g.*, a traffic jam in a major highway at the beginning of a holiday period. In this situation, a flying network of UAVs (FANET) with NFV capabilities can be deployed by a public-safety department of a municipal authority with the aim of improving the situational awareness of the road conditions. As depicted in Figure 6.2, a UAVs infrastructure provider supplies to the municipal authority with a set of these UAVs composing one of the NFVIs introduced in the previous section (see Section 6.2.1). These UAVs build a FANET over the motorway infrastructures, so that, through the execution and interoperation of different softwarization units or VNFs, a network service to monitor the situation on the road is enabled. To this purpose, the municipal authority coordinates the deployment of such service by making use of the NFV MANO stack present in the overall platform. Other UAVs are in charge of collecting relevant information (*e.g.*, video and images). The information produced by the UAVs and the vehicles is delivered to the public-safety department through the mentioned network service, traversing the aerial network comprised by the UAVs and the terrestrial communications infrastructure provided by the Road Side Units (RSUs), facilitating decision-making processes (*e.g.*, the platform could be used to verify that the predictions of traffic flows are fulfilled, and modify the strategies to address the road conditions). In addition, this may complement the resources of cellular access networks serving the users, and thereby preventing a potential stage of congestion caused by dense road traffic situations.

Moreover, the aerial network can be used to support the dissemination of relevant information from the municipal authority to users at cars, making also use of relay VNFs deployed on cars and UAVs. This will enable new types of applications that take advantage of the availability of traffic and driving information, as suggested in [138].

6.3.2. Creating an unheralded vertical service

One of the main benefits of deployments such as the one presented above, is the ability to be adapted in a versatile and expeditious manner in order to address the emerging demands imposed by the altered circumstances. For instance, if an emergency occurs, the depicted deployment could integrate additional UAVs to extend the offered functionality, modify the trajectory of existing ones, execute new functionalities or update the existing ones, and even to release allocated resources to enable the proper operation of the service that will assist to overcome the emergency. In this context, Figure 6.3

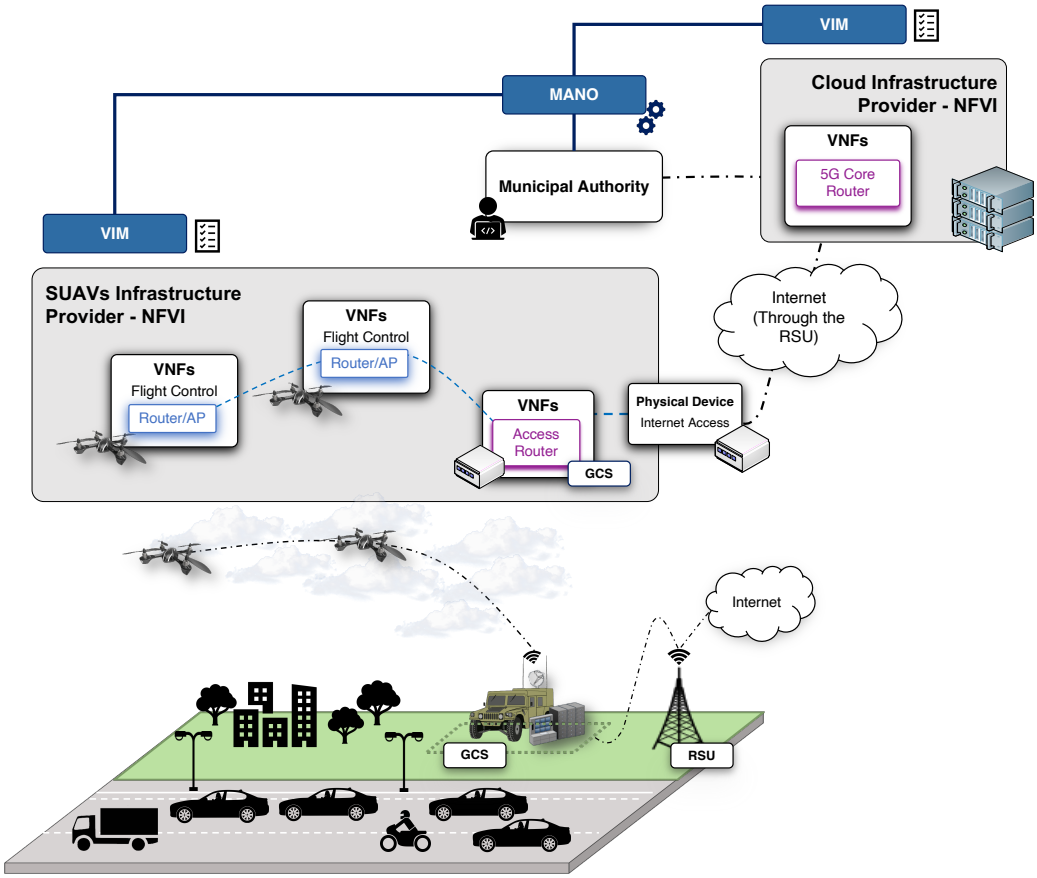


Figure 6.2. *Overview of the initial service provided by the UAVs deployment.*

illustrates a new situation of the presented scenario in which the municipal authority coordinates from the MANO stack the deployment of a complementary service that is aimed to assist the operations during an emergency (*e.g.*, a vehicle collision). In this case, the main purpose is to enable the communications from the UAVs to an emergency response team managed by the department whereas the team is moving towards the location of the emergency. Thus, the response team would have a better situational awareness and prepare and coordinate in advance the steps required to mitigate the emergency. To that end, the UAVs will execute the new functionalities and will be positioned in such a way that they will be able to capture the occurrence of the emergency and to stream the video content to the response team vehicle (*e.g.*, an ambulance). Moreover, the response team members could have access to this situational information due to the functionalities deployed in the vehicle itself in which it travels. To that end, the automotive infrastructure provider provision the municipal authority with the NFVI introduced in Section 6.2.2.

6.3.3. Network service implementation considerations

The following lines aim at outlining the functionality of the proposed network services in order to demonstrate the potential benefits of the framework in a use case such as the previously commented

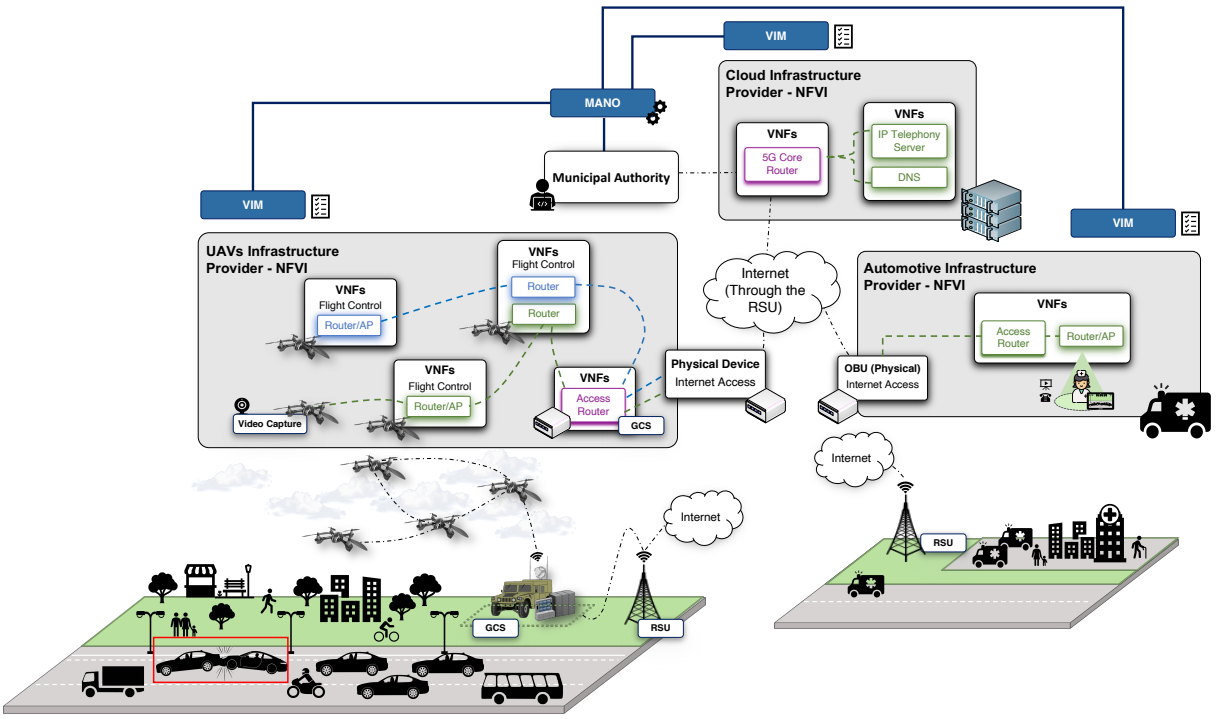


Figure 6.3. *Emergency situation: a complementary network service is deployed to handle the emergency.*

one. To this purpose, it is considered a traffic control situation in which a municipal authority has to extend the service provisioned due to the occurrence of an emergency situation (*e.g.*, a crash between vehicles causing a traffic jam with a high risk of provoking more collisions).

First, it is important to consider that, from the point of view of the municipal authority, one of the design keys to be taken into account when implementing the network service must be the ability to modify the service in real time, and in the most agile and efficient possible manner, in order to address the new requirements that may arise due to a changing event in a scenario such as the one presented here (in which emergency situations can be expected to occur in advance). To this effect, the design of the network service in this experiment is based on a slicing model, where the infrastructure composing the experimental testbed is capable of hosting diverse logical end-to-end networks tailored to fulfil diverse requirements requested by a particular application or service. Furthermore, each slice can be considered as a network service by itself, facilitating the extension of its functionality by the inter-operation with other existing slices. This is possible by means of the use of the NFV technology in each of the infrastructures, allowing the division of the physical resources through virtualization into different logical network components, or VNFs, that will make up each of the slices. This design approach is illustrated in Figure 6.3, representing each of the VNFs composing each of the slices included by the complete service in a different colour. Next, each of these slices are defined along with their purpose.

On the one hand, depicted in the figure using the violet colour, the service encompasses the slice that is in charge of providing connectivity to the rest of the services that are deployed throughout the

municipal authority's platform. This slice, hereafter referred to as the *core-slice*, provides its functionality through the multi-site execution of the VNFs called the *Access router* and the *5G Core Router* on the NFVIs supplied by the UAVs infrastructure and cloud infrastructure providers, respectively. Both VNFs enable a secure transmission of information to the core network hosted by the cloud NFVI over any untrusted access network not defined by the 3rd Generation Partnership Project (3GPP) (*i.e.*, a non-3GPP access network). To do so, the *5G Core Router* implements the user-plane protocol stack defined by 3GPP for a Non-3GPP Inter-Working Function (N3IWF), and supports the network routing functionalities within the cloud domain. Meanwhile, the *Access Router* runs the user-plane protocol stack defined by 3GPP for a 3GPP User Equipment (UE) to reach the core network via an untrusted non-3GPP access. Moreover, this VNF also supports the network routing functionalities to enable the connectivity of the subsequent VNFs and network services accommodated by the UAVs platform with the core network or Internet.

To endow users with an alternative communications channel in traffic jam situation, reducing the overload and congestion of the existing cellular network, the service considers an additional slice represented in the figure with its composing VNFs coloured in blue. This slice, named as *initial-slice*, leverages the slice mentioned before (*i.e.*, the *core-slice*) and addresses a three-fold objective:

1) To supply users situated within the coverage area with an alternative communication channel to browse through the Internet.

2) To support the transmission of information messages from the municipal authority to those who are connected.

3) To enable the delivery of real-time video content to the municipal authority with the goal of improving the situational awareness in the context of high traffic density.

For these objectives, the slice deploys, on each UAV of the UAVs NFVI, a series of lightweight VNFs that can operate with the VNFs of the other UAVs through the FANET presented in Section 6.2.1. These include the *Router/AP*, which provides a Wi-Fi access point to the end-users, besides the network routing functionalities to support the communications of those users. In this slice there are also the VNF called as *Video Capture*, which aims to obtain and disseminate the video content originated by the camera device on-boarded on the UAV to the municipal authority. Finally, this reference use case conceive the instantiation of the so-called *Flight Control* VNFs, which are in charge of both the trajectories and the flight plan that each one of the UAVs hosting its execution must follow.

In the case of an emergency situation, the MANO system included within the framework can coordinate the deployment of an additional slice, which is responsible for providing the appropriate facilities to a response team orchestrated by the municipal authority to deal with the emergency. This slice, referred to as *emergency-slice*, and highlighted in the figure by colouring the VNFs that make it up in green, covers the deployment of an IP telephony service, isolating the VNFs comprising these services from the VNFs of the previously described slice. As a result, the response team can contact the municipal authority, and even receive video content while approaching the accident scene, with

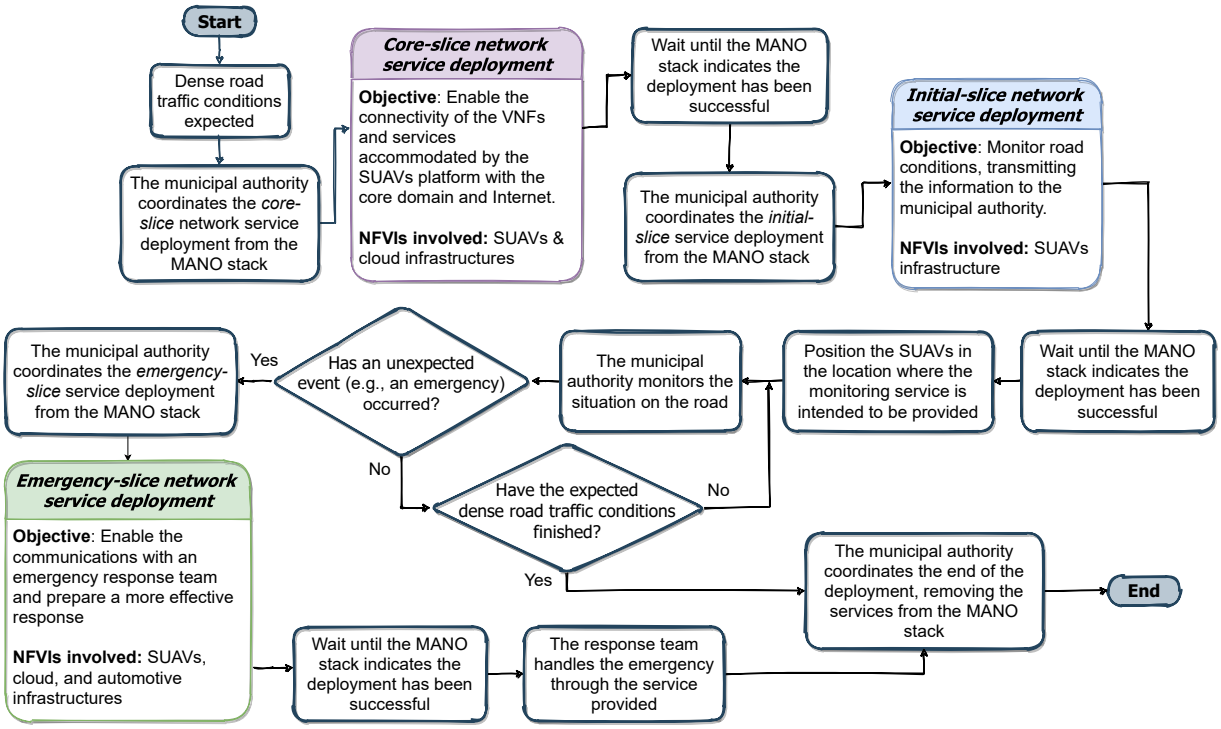


Figure 6.4. *Flowchart of the use case and network service definition.*

the objective of preparing a more effective response to the emergency. In this case, the slice contains two VNFs not introduced before that are called as *IP telephony server* and *DNS server*. These VNFs are responsible for managing the call signalling messages exchanged by the IP phones in order to establish and terminate calls between the response team and the municipal authority, and for providing the name resolution service required by the telephone service, respectively. To this end, this slice carries out the deployment of the VNFs in a multi-site fashion, as shown in the figure, covering the three NFVIs described in the previous section (see Section 6.2). With the aim of emphasizing the major considerations for implementing each network service encompassed by each of the above mentioned slices, Figure 6.4 summarizes graphically with a flowchart the most relevant aspects presented during the use case description.

It is important to emphasize that the slicing basis on which the implementation of the global service has been designed, allows a more flexible orchestration of the services provided, being possible to benefit from the discrimination of the resources made available for each one of the functionalities included by means of every slice. For instance, the complete service is designed to be able to simultaneously provide the described functionality of both the *initial-slice* and the *emergency-slice*, supporting the traffic exchange through the *core-slice*. On the opposite, if the latter undergoes an excessive overload and the emergency needs an increase in the number of reserved resources (for example, including an additional video source to have another perspective of the accident), the *initial-slice* could be removed without affecting the overall functionality of the service managing the emergency.

An additional, relevant aspect to consider within the framework, is the possibility of having periods of intermittent connectivity due to the mobility of the devices utilized, both UAVs and vehicles. In the latter case, as the vehicle moves along the road, the On-Board Unit (OBU) inside the vehicle will actively check for RSUs in range and which one is the best one available to connect to at that time. Once the OBU identifies the one with the strongest signal, and if it was not already connected, it will perform a handover, which in simple terms means that the OBU connects itself to that RSU. During this process, for a few seconds, there is loss of communication with the infrastructure. In a case where a vehicle goes to a place where there is no RSU coverage, the OBU (as well as all the equipment that is connected to it, *i.e.*, the NFVI) will not have a point of connection and until the vehicle returns to a location with coverage, it will have no connection to the vehicular infrastructure. Likewise, this temporary loss of connectivity means that the VNFs that are deployed on the vehicles will not be able to operate with other VNFs deployed on other vehicles or NFV infrastructures, until connection to the infrastructure is re-established. These were the conclusions reached in [134], where different mobility use cases were presented and explored to evaluate the possible effects of connectivity loss within the vehicular infrastructure, and the impact caused on the operation of the VNFs.

Lastly, in the case of the UAVs, considering that it is a platform that it is deployed on-demand in the location considered most appropriate to provide a network service, coordinating and controlling the flight plan (*i.e.*, the trajectories to be followed by the mobile devices) at all times by the infrastructure provider, this intermittent connectivity drawback becomes less significant. Thus, it can be assumed that, by controlling that flight plan, UAVs can be positioned and maintained (*e.g.*, landed on the ground, or in a static hovering situation in the air) in such a way that there is minimal connectivity loss that could affect the operations of the UAVs infrastructure (including the functionality of the hosted VNFs).

6.4. Implementation and Analysis

This section aims at validating the multi-site orchestration framework using the public-safety vertical use case presented in the previous section. First, it presents the experimental testbed used in the evaluation process. It should be highlighted that each infrastructure provider (5G core and UAVs, and automotive) are deployed in different countries, namely Spain, in Madrid (hosted by the Universidad Carlos III de Madrid), and Portugal, in Aveiro (hosted by the Instituto de Telecomunicações). Physically separating both sites brings the setup a step closer to the reality, since the ambulance and the UAVs do not belong to the same network region. Finally, it is important to note that the work of the author of this thesis, in relation to the implementation of the framework, is limited to the context of the Core ands UAVs NFVIs, and to the application of the mechanisms to integrate the Automotive NFVI, which was implemented by the Instituto de Telecomunicações of Aveiro. In any case, the description of the latter NFVI is included below for the sake of completeness.

6.4.1. Experimental testbed

The following lines details the technical implementation aspects of the components comprised within each of the infrastructures included in the testbed that were developed by the author of this thesis. Since the technical aspects of the Automotive infrastructure are not relevant in the context of this thesis, their presentation is omitted. In any case, all the technical details of this infrastructure were carefully described by the Instituto de Telecomunicações of Aveiro, and are available in [137].

5G/Cloud infrastructure provider:

Located in the 5G Telefonica Open Network Innovation Centre (5TONIC) laboratory facilities, the 5G/-cloud infrastructure provider plays a leading role within the proposed framework to enable the provisioning of network services in a vertical such as the automotive industry, utilising UAV devices to support the operations of such services. The design aspects of this infrastructure, aligned with the NFV reference architecture defined by ETSI, and its implementation details (entirely based on open source technologies) have been already presented in the previous chapter (see Chapter 5).

It is worth mentioning that, among the network services that this platform envisages, it includes the development of a 5G core network defined by the 3GPP capable of providing connectivity to end-users in a secure manner, whether or not access comes from a 3GPP access network [116, 117]. In the latter case, a non-3GPP N3IWF would be responsible for providing access to the core network, ensuring confidentiality, integrity and authentication in the course of communications. From this perspective, this NFVI provides the implementation of a basic 5G core network prototype (*i.e.*, there are elements of this 5G core network defined by the 3GPP that are still under development) through the provision of the basic forwarding functionalities defined by the 3GPP for the N3IWF element. Thus, the core network implementation supports the user-plane stack defined by the 3GPP for non-3GPP access networks, making use of the Generic Routing Encapsulation (GRE) [118] and Internet Protocol Security (IPsec) [119] network level protocols with which the 3GPP stipulates this secure access. This enables the connection with the available services that can be offered under the scope of this core network, such as an IP telephony service. In addition, it should be noted that the functionalities described above about 5G core network services have been carried out in the form of VNFs, so that they can be deployed dynamically through the MANO platform integrated in the framework.

Moreover, this infrastructure includes the repository with the implementation of the VNFs that have been presented in Section 6.3.3. These functionalities are an evolution of those presented in the previous chapters, developed by the author of this thesis, and based entirely on open source technologies. In this context, Table 6.1 summarises the most relevant technical implementation aspects of these functionalities, indicating the NFV infrastructures in charge of their execution.

VNF (NFVI)	**Brief Description of Functionality**	**Technical Requirements**	**Featured Software**
5G Core Router (5G/Cloud Infrastructure Provider)	Implementation of the user-plane protocol stack of a 3GPP N3IWF, as well as routing functionalities towards external networks	Prototyped as a VM, using Ubuntu 16.04; 2 vCPUs, 1 GB RAM, 5 GB storage	Linux ip-gre ip-forwarding modules, and the ipsec-tools package
IP Telephony Server (5G/Cloud Infrastructure Provider)	Provide the functions of an IP Telephony service based on the SIP protocol (*i.e.*, proxying of call signalling messages and user registration)	Prototyped as a VM, using Ubuntu 16.04; 1 vCPU, 1 GB RAM, 5 GB storage	Kamailio, an open source SIP server (Linux package)
DNS (5G/Cloud Infrastructure Provider)	Support a name resolution service, to enable user identification in a functional IP telephony service	Prototyped as a VM, using Ubuntu 16.04; 1 vCPU, 1 GB RAM, 5 GB storage	Dnsmasq, an open source DNS server (Linux package)
Access Router (UAVs /Automotive Infrastructure Provider)	Implementation of the user-plane protocol stack of a 3GPP UE, providing access to 5G core network via an untrusted non-3GPP access	Prototyped as a VM, using Ubuntu 16.04; 1 vCPU, 1 GB RAM, 5 GB storage	Linux ip-gre and ip-forwarding modules, and the ipsec-tools package
Router/AP (UAVs /Automotive Infrastructure Provider)	Implementation of a Wi-Fi access point, supporting the assignment of IP addresses using DHCP, and routing functions	Prototyped as LXC container, using Ubuntu 16.04; 1 vCPU, 128 MB RAM, 4 GB storage	Linux ip-forwarding module and isc-dhcp-server package

Table 6.1. *VNFs technical implementation details.*

UAVs infrastructure provider:

As already stated in Section Section 6.3, this part of the platform is responsible to provide a communications service aimed to assist the operations of a public-safety entity (part of a municipal authority) in a context of dense road traffic conditions. Moreover, this platform will undertake the appropriate measures to adapt this service (*e.g.*, change the position of the UAVs , increase the number of UAVs to the mission, etc.) in case an emergency occurs. All the the key aspects related to the implementation of the elements that comprise the NFV UAV-based infrastructure have been presented in Chapter 5, and are omitted in this section so as not to repeat same content within the thesis.

Similarly to the NFVI described in the previous section, the MANO platform can orchestrate the deployment of the functionalities designed for the UAVs Infrastructure Provider environment in order to implement the services defined for each of the slices included in the use-case scenario.

6.4.2. Practical evaluation: deployment times profiling

With the aim of verifying the ability of the experimental platform to adapt to emergency situations, such as the one presented before, this part analyzes the time required in this environment to carry out the deployment of the services that will be executed throughout this use case, especially focusing on

the emergency service.

In this context, the experiments included the measurements of the deployment time that the first slice (*i.e.*, the so-called *core-slice*) takes to deploy the service it comprehends. Accordingly, the deployment was coordinated from the Open Source MANO (OSM) stack, measuring the time it takes to consider the success in completing the deployment. To calculate this deployment time, OSM provides a timestamp corresponding to the instant at which the deployment of the network service was initiated, and another timestamp of the instant when the deployment was completed. In this context, OSM considers that a deployment is completed when all the VNFs composing the network service have been instantiated in their corresponding NFVI, and correctly configured to provide the expected functionality. With this information, by subtracting both timestamps, the time taken for the service to be deployed can be obtained. Once the deployment time is calculated, the network service is removed from the testbed so as to ensure the initial conditions in subsequent deployments. In order to conduct a significant study from a probabilistic perspective, this process was repeated 30 times. It is worth noting that the same conditions were preserved for each of the 30 iterations carried out in the deployment of the service corresponding to the slice. These conditions encompassed the realization of each deployment in a controlled laboratory environment, with the UAVs placed at 5TONIC laboratory (located in Madrid, Spain), and with the vehicle emulating the ambulance by means of an OBU at the laboratory of the Instituto de Telecomunicações (located in Aveiro, Portugal). In addition, both the UAVs and the vehicle were kept in a stationary state (*i.e.*, the UAVs landed in the ground, maintaining the same position, and the vehicle with no movement).

As illustrated in Figure 6.6.a, the values of this experiment have been represented using a box-and-whisker plot, which allows to visualize at a glance the time-series data obtained in relation to the deployment times. Particularly for this slice, it can be seen that the deployment time is gathered around 250 seconds (median value), with 226 and 274 being the values of the lower and upper quartiles (usually identified as Q1 and Q3), respectively. The median value, or also known as Q2 in this type of graph, is represented by a horizontal black line. Using Q1 and Q3, the interquartile range (IQR) can be obtained, which multiplied by the well-known factor 1.5, allows to calculate the maximum length that the whiskers of the representation will have from the values Q1 downwards, and Q3 upwards. Likewise, this allows the identification of outliers (represented by a red cross), which are those values that are beyond the end of the whiskers (both upper and lower).

To determine which is the time required to deploy the second slice, referred to as *initial-slice* in this work, a similar procedure to the one described above was performed from the OSM stack, with the peculiarity that in this case, an instance of the service comprised by the *core-slice* was present in each one of the iterations deploying this new service. Thus, it is possible to study whether the fact of having already a service allocated within the testbed may affect the performance of future deployments. In this case, Figure 6.6.a shows that the deployment time for the 30 runs is between 237 and 304 seconds. This is a very similar result to the one obtained during the previous experimentation block, since both slices have the same number of VNFs composing the service. Considering this, it can also be inferred that the presence of a previous deployment allocated within the testbed does not really affect the

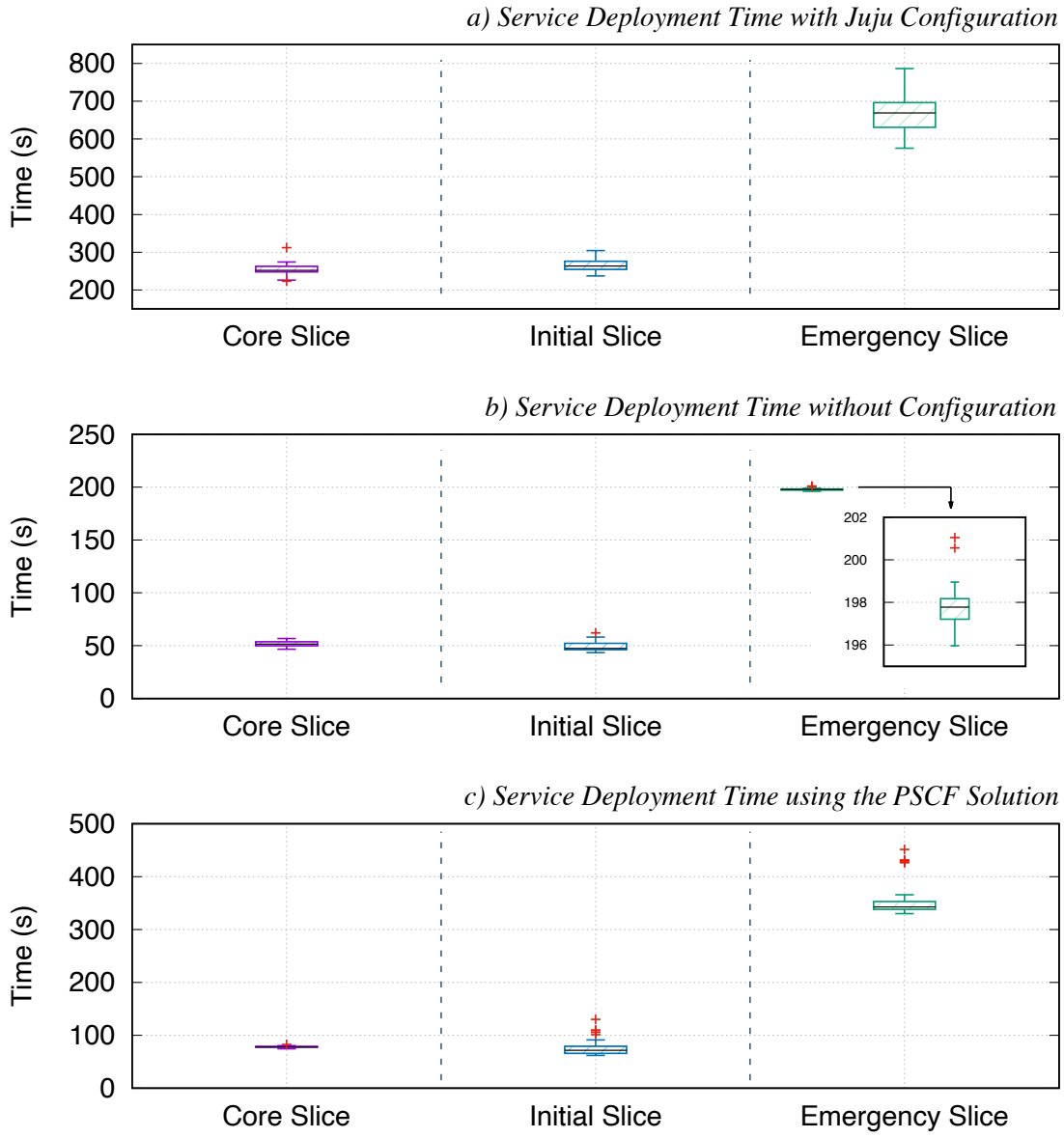


Figure 6.5. *Service deployment time measurements of each slice.*

deployment time of the following services.

Continuing the experimentation process, and using the defined methodology, the next step proceeded to calculate the deployment time of the service encompassed by the slice identified as *emergency-slice*. In this experimentation block, each of the iterations included an instance of the services covered by the *core-slice* and the *initial-slice* before proceeding to the deployment time profiling of the *emergency-slice*. Figure 6.6.a in this case shows a noticeable increase in relation to the deployment time compared to the previous slices. In particular, the time to deploy the service in charge of addressing the emergency situation has a median value of 668 seconds, with a maximum spread in

between 575 and 780 seconds. In other words, it can be seen that the service may even take around 13 minutes to be deployed. Considering that both UAVs and vehicle NFVIs have similar accessible resources, and the provision of services using a set of standardized operations through the utilization of NFV technologies in the framework, this increase is due to the fact that the number of VNFs included in the *emergency-slice* is larger (specifically, the service has six VNFs against the two included in each of the other slices).

Although the deployment time may seem reasonable in terms of comparison with the reference values considered in 5G [3], the upward trend observed during the experiments due to the number of VNFs involved in the service, leads to consider if the proposed framework has the sufficient capability to agilely and flexibly be adapted in scenarios where the service requires greater complexity, with a larger number of VNFs, to manage an unexpected situation. Indeed, there are emergency response teams such as SAMUR (the municipal emergency service in Madrid) which have an average emergency response time of about nine and a half minutes [139], what indicates that if a service such as the one proposed here were intended to be deployed, the team would have to initiate the emergency coordination without the support of the designed communications service.

With the objective of identifying at what stage of the deployment this time can be reduced, the analysis carefully studied how the MANO platform operates throughout the instantiation of a network service. Before this, it is important to remark that the MANO platform within the framework is based on the OSM Release SEVEN (see Section 6.4.1), which implements the different components of a MANO platform using a cloud-native model based on Docker containers, and provides a Kafka bus to enable the interaction of the different elements/containers with each other in order to automate the deployment of end-to-end network services. In this respect, once the instantiation process is executed, OSM consecutively undertakes the following phases:

1) Processing of information uploaded to the MANO platform with both the Network Service Descriptor (NSD) and the descriptors of the virtualised functions (VNF Descriptors, or VNFD) that comprise it. In this first phase, the MANO platform determines which ones of the VNFs require configuration by the VNFM element included in OSM, implemented through Juju.

2) Coordination with the different VIM entities configured in the MANO platform responsible for managing and allocating the resources that will be used by the VNFs in each of the corresponding infrastructures where they are meant to be executed. Simultaneously, the MANO platform at this stage reports Juju to start the preparation of the VNF configuration environment, and to this purpose, Juju runs a series of Linux containers within the host in charge of executing OSM. In this context, a container is created for every VNF that require configuration, associates itself with one of those VNFs, and Juju carries out the appropriate software installation to enable the subsequent configuration of the associated VNF. Moreover, this is possible through the execution of a set of Juju scripts called proxy charms, which enable the use of the Ansible playbooks technology by means of the base charm layer contributed within the OSM software [85] (all details of this contribution were described in Chapter 3).

3) Once OSM is notified by the VIMs about the proper instantiation of each image of every VNF (*i.e.*, the softwarization unit loaded into the VIM with pending configuration to provide specific functionality) has been correctly carried out, it also collects the information regarding the management IP address that allows the configuration of these VNFs. Next, OSM transmits this information to Juju to proceed with the configuration tasks, executing the specified actions for each VNF in their corresponding container.

4) Finally, when each container completes the configuration activity of its associated VNF, Juju sends a message to OSM to inform that the configuration process has been completed. Once OSM processes this message, the deployment of the service is considered complete.

To ensure that this process is as efficient as possible, OSM processes in parallel both the instantiation and configuration stages defined for phases 2 and 3. Despite this, as observed in the previous chapter of this thesis (see Chapter 5), this process requires such a high processing load that it results in a significant delay to the service deployment time. Furthermore, it can also be appreciated how this delay is also accentuated by the complexity of the synchronisation tasks between Juju, the Linux containers deployed by the own Juju, and OSM to carry out the configuration and the lifecycle management of every instantiated VNF. In the light of the lessons learned from the previous chapter, and in order to confirm that the configuration stage is a possible candidate for improvement to reduce the service deployment times, the next experiments measure the time taken to deploy the services contained in each one of the slices, but this time without configuration. To do this, the same methodology mentioned above was repeated, consecutively instantiating each service without configuration 30 times. In this case, Figure 6.6.b depicts how these times are indeed far below from what was seen in the previous set of tests, and how the VNF configuration increases the deployment time by, at the very least, a factor of 3. The following section elaborates on the development carried out in the scope of this work to mitigate this factor within the proposed framework.

6.4.3. Publish–Subscribe Configuration Function

In order to overtake the previously analyzed limitation in terms of service deployment time, this part presents the developed and integrated solution within the overall experimental platform. This solution intends to to carry out a distributed implementation of the functionality specified for the VNFM component of the ETSI NFV architectural framework.

One of the design keys of this solution is that it considers the decentralization of the VNF configuration function so that it does not result in a processing overload for the host executing the MANO stack (in this case, the machine hosting the OSM software). In particular, the solution proposes to localize an instance of the novel VNFM implementation in each one of the NFVIs integrated in the MANO stack, and to place it close to the VIM in charge of the resources of this infrastructure. In this context, each instance will be only and exclusively responsible for the configuration of those VNFs allocated by the infrastructure where it is located.

An additional key aspect to consider about this solution is that it is based on a publish–subscribe model. The main attribute of this model, which is widely spread in the Internet of Things (IoT) environments [140–143], is that it allows the exchange of information related to events in an efficient and asynchronous manner. To that end, the model defines topics based on the use of identifiers, that will be in charge of categorizing and organizing the information flow related to different events. With these identifiers, the elements called within the model as publishers can produce or publish information referring to a specific event through its predefined topic. Subsequently, this information will be consumed by those elements subscribed to the topic under consideration. To enable both, the model also defines the figure of the broker, which is in charge of receiving the information published under a topic, storing it in a persistent manner, and asynchronously distributing it to subscribers.

Based on this perspective of a publish–subscribe model, the solution, hereafter referred to as the Publish–Subscribe Configuration Function (PSCF), proposes the use of this model to synchronise the different elements that come into play during the configuration stage of a network service and to coordinate the configuration activities. To do so, an initial information flow was defined to be published at the time of deployment by the MANO stack, containing the configuration data of each of the VNFs comprised by the network service. This first information flow is also structured in groups in which the information contained corresponds to the VNFs that are going to be deployed over the same infrastructure, using for this purpose a topic per infrastructure when publishing (for instance, 'vnfs_information_nfvi_x'). Thus, each PSCF could subscribe to the appropriate topic and only receive the precise information for the configuration of the VNFs under its charge (*i.e.*, those allocated within its particular NFVI). In addition, the data published under a topic like the one previously mentioned, includes the VNF identifier (for example 'id = vnf_1') and the set of configuration actions for the associated VNF, so that the PSCF is able to process and collect the configuration information to proceed with a subsequent VNF configuration. On the other hand, once the PSCF has the necessary information to carry out the configuration of the VNFs, it has to realize when these VNFs are ready to be configured. To this purpose, a new information flow was defined to be shared following the publish–subscribe model, in which each VNF publishes when it is active (*i.e.*, when it has been completely instantiated). Again, a topic has been defined for each infrastructure (*e.g.*, 'vnf_ready_nfvi_x'), including as data under the publication the VNF identifier, along with the management IP address that enables the communications between the VNF and the PSCF in order to receive the configuration commands (*e.g.*, 'vnf-id: 1; ip-address:10.4.16.10'). This information can be consumed by the PSCF after prior subscription to the topic, allowing to start the configuration actions defined by the first information flow, and thus carry out the configuration assignment. Figure 6.6 summarizes this synchronization process to perform the VNF configuration proposed by the PSCF solution.

To carry out the implementation of an initial version of the PSCF and validate the viability of the proposed solution, a virtual machine was configured within the segment of the experimental testbed denominated as 5G/Cloud infrastructure provider, installing the necessary software to develop the required functionalities within the publish–subscribe model, as well as those required to carry out the configuration of the VNFs. On the one hand, the machine is supplied with the Kafka open source software [144], which allows the streaming of events based on the publish–subscribe model, imple-

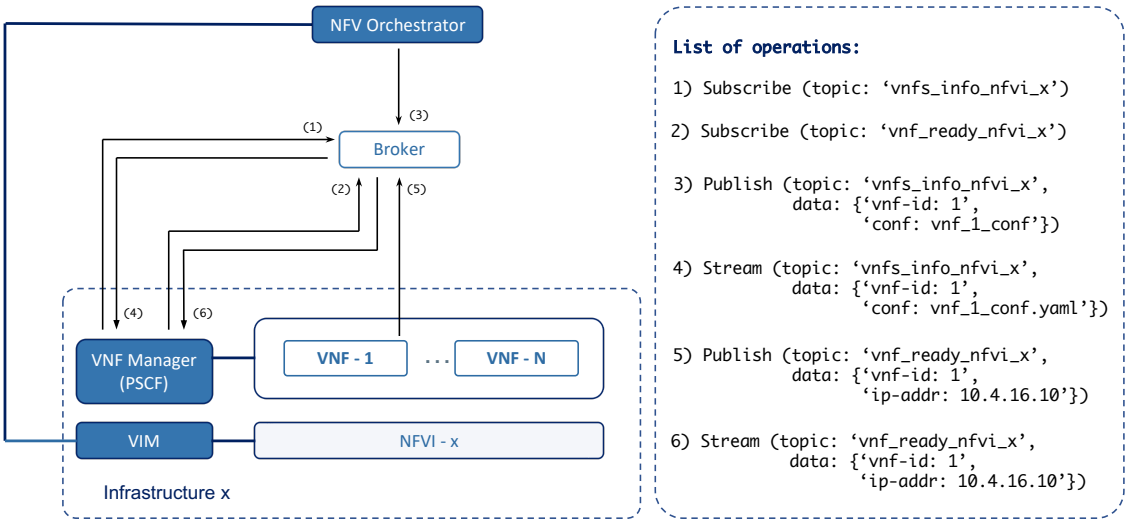


Figure 6.6. *Flow-operations diagram of the Publish–Subscribe Configuration Function.*

menting the role defined within the model for the figure of the broker. This software enables a communications bus, denoted as Kafka bus, that receives the information flow of the events produced by the publishers (organised as mentioned above by means of topics), and makes it available to those consumers that are subscribed. Additionally, this machine incorporates the python library kafka-python, implementing a client of the Application Programming Interface (API) provided by the Kafka to interact with its bus. This library is used in the PSCF for the creation of a script that allows to consume the messages published on the bus, processing them to gather the pertinent information, and executing the configuration commands of each VNF in parallel (once the message corresponding to the completion instantiation VNF has been received). This kafka-python library is also installed at the host executing the MANO stack (*i.e.*, the machine running the OSM software), so that it can publish the information related to the VNFs requiring configuration once it carries out the deployment of a service.

On the other hand, to carry out the configuration function of the PSCF once the corresponding message has been consumed and processed, the machine has been provided with the installation of an Ansible server [84], which is a technology that allows the automated provisioning of software and its configuration management for deploying applications. Furthermore, Ansible provides a functionality that makes possible to describe and group the sequence of configuration actions for one or more machines to be carried out by the Ansible server through what is known as Ansible playbooks. These playbooks are specified in *.yaml* files so that they can be reused as often as necessary. In this solution, each VNF has an associated playbook that is loaded in the OSM platform through the VNFDs. These VNF playbooks conform the first information flow that has been defined to be streamed through the publish–subscribe model of PSCF solution. To simplify this process, and thus avoid altering the OSM software stack, the playbooks of each VNF have been manually pre-loaded in the PSCF machine. In this manner, the PSCF will proceed to execute these playbooks (as mentioned above, in parallel), once it consumes the message published by each VNF specifying that it is ready. In this sense, each VNF in-

cludes a python script that is executed in the machine startup, and with which, through the mentioned library kafka-python, publishes the message in the Kafka bus reporting that it has been instantiated. This, along with the script processing the published messages, are available as open source code at [145].

With this initial version of the PSCF solution, the analysis included the deployment of the services contained by each slice, and measured the time it takes to complete the whole process (*i.e.*, instantiation plus configuration). As it can be seen in Figure 6.6.c, this time is quite lower than the one obtained in the previous scenario with the configuration made by OSM through Juju. This confirms the viability of the PSCF solution. In fact, these results were collected by simply having one instance of PSCF. Hence, it can be perfectly assumed that these values can even be better by distributing one instance of the PSCF per each NFVI, as indicated in the design keys of the solution.

6.4.4. Video service

The next objective within the experimentation was focus on corroborating that the overall service can offer the expected functionality by means of the interoperation between the three slices. To do this, this experimental deployment was done using the development of the PSCF for configuring each VNF. Thus, in the case of an emergency (*e.g.*, a collision between vehicles), the municipal authority can become aware of the emergency and effectively coordinate the required operations for handling the emergency together with the public-safety department. For this, the use case contemplates the option of enabling the simultaneous streaming of different real-time video feeds. In particular, the first video feed enables to monitor a section of road (known in advance as a risk or conflict zone), improving the situational awareness on the part of the corresponding municipal authority. From this video content, the municipal authority detects that an emergency has occurred and the scenario includes an additional real-time video feed to capture in detail the occurrence of the emergency. Then, this additional video content is streamed to the response team that is on its way, to allow the definition of an action plan, prioritising the most urgent actions.

In this context, once the services of the three slices were deployed, and correctly configured, an evaluation of the network performance supported by the interoperation of these services was carried out with the aim of proving the stable and flawless operation of the video service explained. On the one hand, the network performance available for the video content of the *initial-slice* service has been evaluated. This video content will be retransmitted from the VNF offering a wireless access point allocated by the UAVs NFVI, to the municipal authority (see Figure Figure 6.3). In order to analyze whether the network performance can support that video content, the available bandwidth between the mentioned endpoints has been measured using the Iperf tool [146], and the Round Trip Time (RTT) with the Ping tool [147]. The evaluation results for this scenario indicate an available bandwidth of 21.20 Mbps, and an RTT of about 7 ms. From the latter value, it can be inferred that the end-to-end delay performance is about 3.5 ms. These outcomes comfortably comply with the requirements defined by some relevant organizations within the telecommunications sector for this type of real-time video services. In particular, the ITU-T recommends end-to-end delays below 150 ms so that the display at

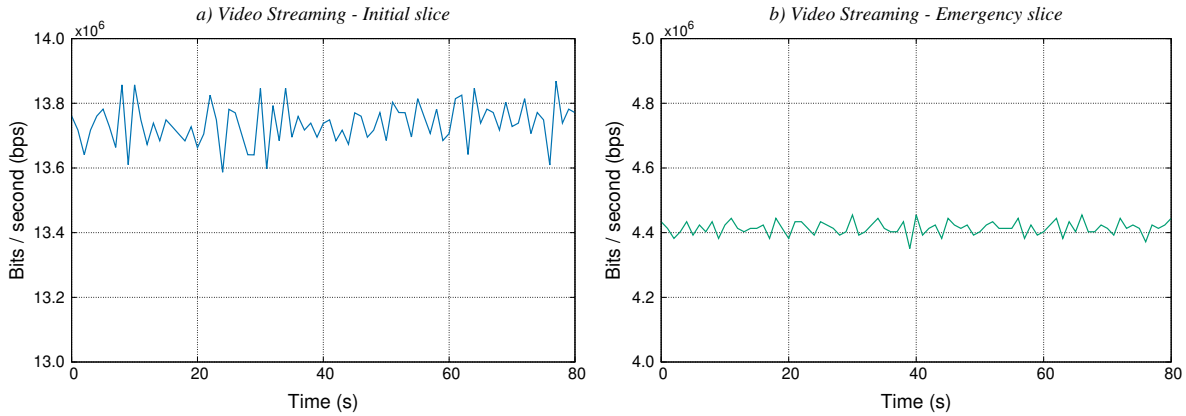


Figure 6.7. *Traffic flows of the video service.*

the destination is not significantly affected [120]. Moreover, the 3GPP technical specification [148] in charge of defining the Key Performance Indicators (KPIs) for the UAVs operations within the 5G communications systems, indicates that the available bandwidth during the streaming of real-time video must be at least 0.06 Mbps, and that the end-to-end delay cannot exceed 100 ms. As can be noted, this specification is even more restrictive in terms of end-to-end delays than the ITU-T recommendation. In addition to this, the obtained results also manifest that the transmission of a high-definition (HD) quality video content (which requires a bandwidth of at least 5 Mbps for a resolution of 720p) is feasible.

With respect to the emergency scenario, the same analysis was performed, but this time between the endpoints that allow the video content streaming with the purpose of managing the emergency situation, *i.e.*, between the VNF providing an access point allocated for the *emergency-slice* by the UAVs NFVI, and the wireless access point that will be used by the response team, allocated within the ambulance vehicle of the automotive NFVI (both functionalities are part of the service implemented by the *emergency-slice*). As in the previous scenario, notwithstanding that the results are limited by the wireless communications technology of the automotive platform (*i.e.*, WAVE/IEEE 802.11p), the requirements mentioned above for supporting real-time video streaming are sufficiently fulfilled since the results provide an available bandwidth of 6.85 Mbps, and an RTT of almost 40 ms (or in other words, the maximum end-to-end delay never exceeds the 20 ms). In the case of the latter, the value is increased with respect to the previous scenario, since the points used for the aforementioned measurement are located in different geographical points. In fact, they are in different countries: the former is in Madrid, Spain, and the latter in Aveiro, Portugal.

Given that the network performance evaluation has demonstrated the feasibility of supporting the video service, the outlined real-time videos are streamed. To do so, the Iperf tool has been used again, since it allows to transmit an UDP traffic stream at a specific data-rate to emulate the video content stream. As can be seen in Figure 6.7, with respect to the first video content, a rate of around 13 Mbps has been selected to emulate a very high-definition quality video that allows a wide stretch

of road to be monitored in the finest possible detail. According to the second video content, a high-definition quality video has been emulated, lower than the previous one, so that it can be transmitted to the response team's ambulance. Furthermore, it can be seen that the service offered supports the simultaneous transmission of both contents.

6.5. Conclusions

With the aim of exploring the potential of using the NFV system based UAVs to support the deployment of telecommunications and vertical services in resource-constrained situations, this chapter has analyzed the portability of that system to a different environment, where resource-constrained hardware platforms might be available. In particular, the environment considered was the existing research development conducted by the Instituto de Telecomunicações of Aveiro, which employs an NFV infrastructure based on a fleet of vehicles. As a result of the collaboration with this institute, this chapter has presented a novel framework that considers the joint integration of three types of NFV infrastructures, with the aim of creating a distributed and more complete NFV ecosystem. Thus, supporting the flexible deployment of vertical services in situations where there is noticeable resource constraints. In this context, this chapter has presented the following main contributions:

- Definition of an NFV framework capable of integrating aerial and vehicular NFV infrastructures, to enable the cost-effective and flexible deployment of vertical services in resource-constrained situations.
- Description in detail of the realization of a public-safety vertical use case, to emphasize the practicality and potential benefits of the proposed framework.
- Integration of two remote NFV infrastructures: an infrastructure of UAVs, which may be deployed on demand, and an automotive infrastructure, supporting the opportunistic provision of services. This latter infrastructure, provided by the Instituto de Telecomunicações of Aveiro.
- Contemplate the network slicing model as a design key, to exploit the ability of modifying the services in real time, in the most agile and efficient possible manner.
- Description of the implementation details of both the framework and the network services, through the use of open source technologies.
- Development of a novel solution based on the publish–subscribe model to agilely carry out the configuration of VNFs, significantly reducing the deployment times in an emergency scenario. The source code of this solution is available at [145].

The approach followed in this and previous chapters has demonstrated its ability to flexibly deploy virtualized functions in different computing domains (*e.g.*, cloud/edge, UAVs and/or vehicular infrastructures) to realize the execution of moderately complex multi-site services. In this context,

this approach has relied on typical network layer routing mechanisms (*i.e.*, routing over the Internet) and/or the use of overlay network technologies, such as Virtual Private Networks (VPNs) to provide connectivity among VNFs that are executed in diverse domains, distributed in different geographic locations. However, these mechanisms have significant limitations to support data communications among VNFs deployed in different domains. For instance, using the final scenario of the use case presented in this chapter as a reference, the use of network-level mechanisms hampers the proper isolation between the service offered with *initial-slice* and the service provided with the *emergency-slice*. To address these limitations, the following chapter is focused on the inter-site communications, exploring appropriate mechanisms to enable the exchange of data traffic among VNFs that are located in different NFV domains.

CHAPTER 7

A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services

As already discussed in the different chapters of this thesis, Network Functions Virtualization (NFV) is a key technology for network automation and is being instrumental to materialize the disruptive view of 5^{th} Generation of Mobile Networks (5G) and beyond mobile networks. In particular, 5G embraces NFV to support the automated and agile provision of telecommunication and vertical services as a composition of versatile virtualized components, referred to as Virtualized Network Functions (VNFs). It provides a high degree of flexibility in placing these components on distributed NFV infrastructures (*e.g.*, at the network edge, close to end users). Still, this flexibility creates new challenges in terms of VNF connectivity. Typically, the usual approach adopted for this class of links has been to rely on the best-effort inherent to IP networks, assuming that these networks provide the best possible performance.

To address these challenges, and with the aim of accomplishing the Objective O4 defined within the scope of this thesis, this chapter introduces a novel secure link-layer connectivity platform, L2S. This L2S solution can automatically be deployed and configured as a regular multi-site NFV service, providing the abstraction of a layer-2 switch that offers link-layer connectivity to VNFs deployed on remote NFV sites. With this, inter-site communications can be effectively protected using existing security solutions and protocols, such as IP security (IPsec). Thus, the L2S solution supports the creation of secure data-link layer overlay networks over the IP networks, and enables the exploitation of these advanced links by means of an Software Defined Networking (SDN) framework. In this sense, the SDN framework embraces traffic management mechanisms that may be based in traffic engineering tech-

niques to efficiently decide the optimal manner to manage and orchestrate the inter-site links, as well as their transit traffic.

For this, Section 7.2 introduces the conceptual design of the L2S platform, covering deployment, and configuration aspects. Based on these conceptual design details, Section 7.3 presents the prototype implementation based on open source software technologies of L2S, and its functional validation by means of using this solution to support a multicast-based IP television service. After this, Section 7.4 analyzes the integration of an SDN framework to facilitate the orchestration of inter-domain point-to-point links that can be provided by the L2S solution. Finally, Section 7.5 presents the major conclusions of this chapter, summarizing the contributions achieved.

7.1. Introduction

Undoubtedly, the development of next-generation mobile networks, and in particular the recently available 5^{th} Generation of Mobile Networks, or 5G, has revolutionized the landscape of broadband wireless access connectivity and mobile communication services. On the one hand, 5G significantly increases the performance of user communications at a large scale, in terms of bandwidth, latency, resiliency, and availability. On the other hand, the development of 5G has involved key vertical industries and stakeholders from the early design phases, promoting not only technological but also business innovation. This has led to an unprecedented evolution of public and private mobile networks, fostered by the groundbreaking vision of an extremely flexible communications infrastructure, capable of: *i)* integrating multiple and geographically distributed compute, storage, and network resources, owned by diverse telecommunication operators, infrastructure providers, and other 5G stakeholders; and *ii)* accommodating highly heterogeneous and changing service demands and use cases from different vertical sectors (*e.g.*, automotive, manufacturing, public-safety, smart cities, etc.).

At the heart of 5G, softwarization plays a fundamental role [6], supporting the replacement of traditional specialized hardware equipment by versatile software components. In this respect, 5G adopts the sEuropean Telecommunications Standards Institute (ETSI) Network Functions Virtualization (NFV) [7]. ETSI NFV provides a reference architectural framework to automate the management and orchestration procedures of telecommunication and vertical services. These services are built as a composition of network functions, provisioned as software appliances running on virtual representations of hardware equipment (*e.g.*, virtual machines or virtualization containers). This way, NFV enables the automated deployment of 5G services as connected graphs of Virtualized Network Functions, or VNFs. From a management perspective, NFV technologies alleviate the dependency of service provisioning on specialized hardware, as multiple VNFs can be executed on more generic virtualization-capable server computers with diverse capacities. Still, VNFs can take advantage of specific hardware features if they are available, *e.g.*, encryption acceleration, allocation of physical Central Processing Units (CPUs) to applications, provision of direct access to network interface cards, etc. The virtualization of network functions and their automated management help reducing capital

and operating expenditures in the provision of 5G services. In addition, the use of standard interfaces opens the market of virtual network functions to new vendors and software developers, favoring the availability of a wider and potentially open catalogue of network functions and added-value NFV services.

It is worth highlighting that virtualization implies a high degree of flexibility to place network functions at different locations, provided that there are sufficient computing, storage and networking resources to support their proper operation. This creates new collaboration and business opportunities for telecommunication operators, infrastructure providers, service providers, and other 5G stakeholders, which may result in the deployment of telecommunication and/or vertical services across multiple sites hosting NFV infrastructures.

Nevertheless, this flexibility regarding VNF placement opens new challenges in terms of their connectivity. Ideally, the multi-site nature of an NFV ecosystem should be opaque to the stakeholders requesting a service deployment, as it should be provisioned in functional terms, independently of whether the instance of the service has been deployed at a single NFV site or several of them. However, whereas the connectivity of VNFs at a single NFV infrastructure can easily be supported with the creation of virtual local links, this is problematic in a multi-site NFV ecosystem. This is because NFV sites may be geographically distributed and interconnected through untrusted network domains owned by multiple Internet service providers that in most cases, must remain oblivious to NFV operations. Moreover, sites may belong to different stakeholders and be subject to distinct management and orchestration policies, enforced by a variety of mechanisms.

For these reasons, inter-site communications in distributed NFV ecosystems have commonly relied on existing layer-3 inter-site routing mechanisms (*i.e.*, Internet routing) and/or on the use of overlay network technologies, such as Virtual Private Networks (VPNs). This approach has satisfactorily been used in different research projects funded by the European Union, with the goal of implementing distributed 5G testing facilities across Europe, fostering extensive trials and demonstration with 5G technologies that involve vertical industries and key stakeholders. In particular, the 5GinFIRE project built an experimentation ecosystem with nine NFV testing facilities, spanning Europe and Brazil [149]. In this ecosystem, data-plane communications among VNFs deployed at different sites were supported through layer-3 routing and a VPN-based overlay network (see Chapter 3 of this thesis). Following a similar approach, the 5G-VINNI project leverages layer-3 routing and VPN functionalities to create dedicated logical networks that enable the interconnection of VNFs running on a set of eight 5G facilities [150]. The 5G-EVE project defines a specific functional entity (the Data-Plane Network Gateway) that is deployed on each of the eight 5G facilities involved in the project [151]. These data-plane gateways provide secure data connectivity among the different sites, using IP tunnels (based on IP security, IPsec [119], or Generic Routing Encapsulation, GRE [118]) and layer-3 routing. This way, the 5G-EVE project supports layer-3 connectivity among remote VNFs.

Although this approach has proven to be effective to enable secure layer-3 connectivity among distant NFV sites, it still presents non-negligible limitations to support data communications among VNFs deployed at those sites, considering: *i)* the bounded capacity of layer-3 routing mechanisms

to guarantee the isolation among multi-site NFV services; *ii)* the potential need for undesirable day-2 configurations for the VNFs themselves, making the multi-site nature of the NFV ecosystem not oblivious to tenants; and *iii)* the potential requirement to configure additional forwarding state on the underlying network infrastructures that support inter-site and inter-VNF connectivity.

7.2. Description of the L2S Platform

Based on the discussion about the existing models to support data communications among VNFs deployed at different NFV sites (*i.e.*, inter-site communications) described in [152], this section is devoted to presenting the design of the L2S platform. This platform intends to provide secure link-layer inter-site communications within a multi-site NFV ecosystem where several sites can be used to automatically deploy network services.

Figure 7.1 outlines the conceptual design of the L2S platform. The platform provides the abstraction of a layer-2 switch, supporting link-layer connectivity among VNFs deployed on different NFV Infrastructure (NFVI)s. This layer-2 switch operates as a multi-site network service, with an L2S VNF on every NFVI. Therefore, the deployment of the L2S platform can be automatically provisioned as any other network service over the NFV infrastructures, without requiring the installation of additional network equipment at the NFV sites.

The layer-2 switch implemented by the L2S platform presents several access ports at each NFV infrastructure, which can be assigned to different VLANs. Thus, the access ports associated to the same VLAN can exchange layer-2 frames through the platform. To this purpose, an L2S VNF encapsulates the layer-2 frames that receives on each of its access ports into an IP tunnel, which delivers this traffic to every other L2S VNF that has an access port in the same VLAN. These IP tunnels can be implemented existing standard protocols, such as VXLAN [89] (the option chosen for implementation of the L2S platform) or GRE [118]. Moreover, since the L2S VNFs manage the traffic entering and leaving the NFV infrastructure, they represent an appropriate location to enforce security policies and protect inter-site VNF communications. In this contexti, the design of L2S relies on existing security solutions to protect the information transmitted within IP tunnels among L2S VNFs, and provide guarantees on confidentiality, integrity, authentication, and non-repudiation. Security solutions could be used at different layers of the TCP/IP protocol stack, *e.g.*, IPsec [119] (the technology used for the implementation), MACsec [153], or layer-2/layer-3 VPNs [154, 155]. This way, inter-site VNF communications are securely provisioned across any potentially untrusted Internet service provider networks that interconnect the NFV sites.

7.2.1. Deployment and configuration aspects

Regarding the deployment of the L2S platform, it is assumed that each site supports external communications following a layer-3 approach (*i.e.*, VNFs can gain access to external networks via an edge router at every site). In addition, a number of virtual links exist at each NFVI (they can be pre-created

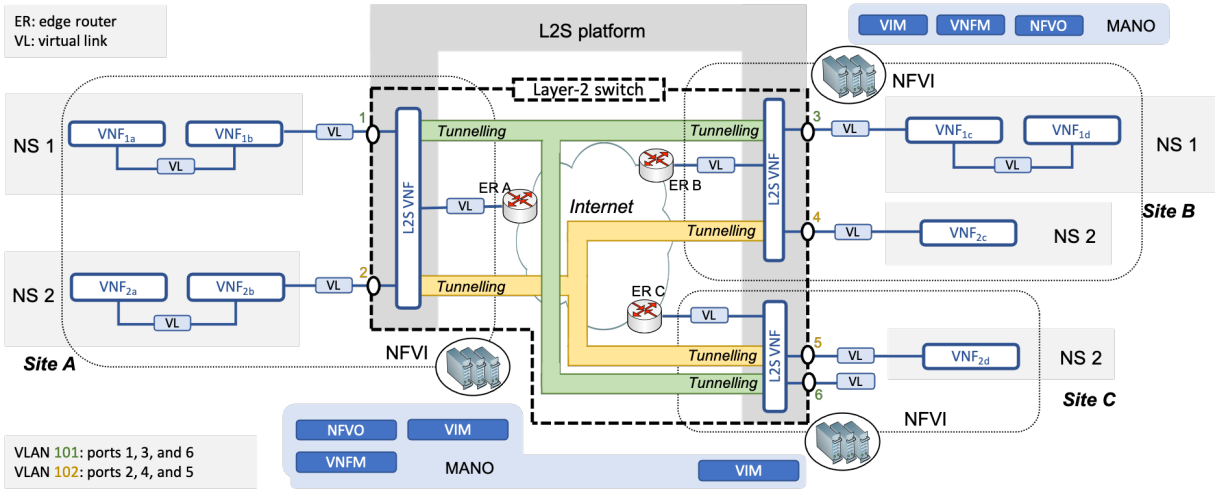


Figure 7.1. *Conceptual design of the L2S platform.*

using the Virtualized Infrastructure Manager (VIM) that manages the NFVI resources). These virtual links will be used to support the inter-site communications of the VNFs running on the NFVIs. The deployment of the L2S platform as a multi-site network service results in the instantiation and configuration of an L2S VNF at every NFVI. Upon instantiation, each L2S VNF is connected to the virtual links pre-created at its NFVI, as well as to a virtual link providing connectivity towards an edge router of the site.

Following the ETSI NFV standards [156], each L2S VNF will automatically be configured through the VNF Manager (VNFM) associated with it. This configuration includes the assignment of access ports to VLANs at the VNF, as well as the creation of IP tunnels towards other L2S VNFs. The specific day-1 configurations to be done on each L2S VNFs will depend on service-level agreements established by infrastructure providers. These agreements will determine the configuration parameters for any L2S VNFs, including VLAN identifiers, network addresses allocated to IP tunnel endpoints, and cryptographic keys to initialize the security mechanisms that will protect IP tunnels. In addition, providers may make unilateral agreements with other providers to configure specific VLANs for the sole purpose of their inter-site communications. These VLANs would be enabled on the layer-2 switch provided by the L2S platform, resulting in the configuration of different VLANs at each L2S VNF.

7.3. Implementation and Validation

As suggested by the previously described conceptual design, one distinctive property of the L2S platform is its potential for innovation using existing technologies and protocols. To verify this property, the following lines present the prototype of a functional L2S VNF using standard Internet protocols and open source technologies.

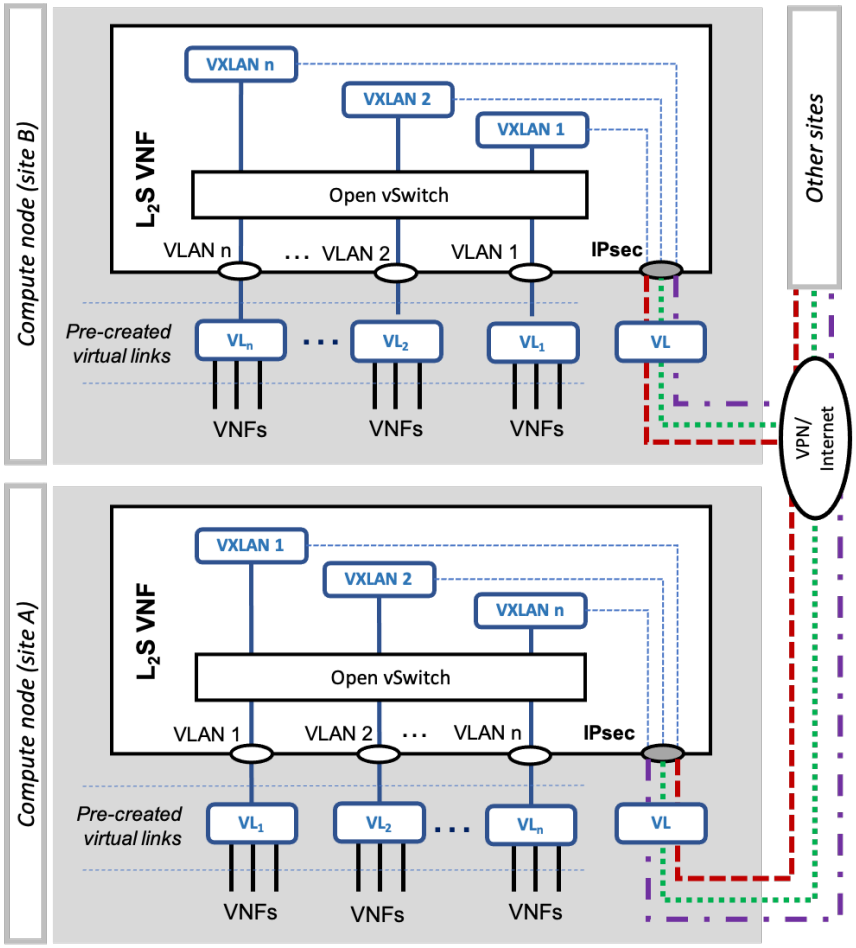


Figure 7.2. *Implementation of the L2S VNF.*

7.3.1. Implementation of the L2S VNF

The implementation of the L2S VNF is outlined in Figure 7.2. This component is based on *Open vSwitch*, an open source, programmable, production-quality virtual switch [157]. As commented in the previous section, the solution assumes that several virtual links are pre-created at each NFVI for the purposes of inter-site communications. Upon instantiation, an L2S VNF is provisioned with several virtual interfaces, each one attached to one of these virtual links. In addition, an *Open vSwitch* instance is created on the VNF. Every interface of the L2S VNF that connects to a pre-created virtual link is then added as an access port to the *Open vSwitch* instance and assigned to a VLAN. For each VLAN, a VXLAN interface is also created and attached to the *Open vSwitch* instance. This interface behaves as a VXLAN tunnel endpoint [89], implementing an IP tunnel towards every other L2S VNF that has an access port on the same VLAN. Traffic encapsulated within every IP tunnel is protected through IPsec transport mode. To this purpose, the implementation uses *strongSwan*, an open source IPsec implementation for Linux [158].

The L2S VNF was prototyped in a virtual machine with Linux Ubuntu 16.04.5 LTS operating sys-

tem, with 1 vCPU of processing, 1 GB RAM, and 20 GB of storage. A *bash* script on the virtual machine supports the automated configuration of the L2S VNF, including: *i)* the creation of the *Open vSwitch* instance on the virtual machine; *ii)* the configuration of the VLANs and the VXLAN interfaces; and *iii)* the establishment of IPsec security associations to protect IP tunnels. The *bash* script works with a configuration file, containing the parameters that are needed to adapt the L2S VNF to different deployment scenarios. These parameters include VLAN and VXLAN identifiers, IP addresses of other L2S VNFs (that is, IP tunnel endpoints), and cryptographic keys to initialize IPsec procedures. Configuration parameters would be provided to the Management & Orchestration (MANO) system involved in the deployment of an L2S VNFs, which would then use them along with the *bash* script to instantiate the VNF.

7.3.2. Performance evaluation

The several practical experiments conducted to gain a better understanding about the performance aspects of the solution are covered herein. In particular, these experiments include the deployment of a network service consisting of two communication endpoints (see Figure 7.3). Each of them was connected to an instance of an L2S VNF, which was in turn attached to an edge router. Both edge routers were provided with a back-to-back connection. The communication endpoints and the edge routers were prototyped as Linux virtual machines (Ubuntu 16.04.5 LTS). Regarding the L2S VNFs, the experiments used the implementation described in Section 7.3.1. The network service was deployed as a set of interconnected virtual machines on a single server computer with sufficient resources, to preserve the results of the performance evaluation from external interference. After the deployment, the L2S VNFs were configured to connect both communication endpoints to the same VLAN. This way, both endpoints were interconnected at layer-2 through the L2S platform, being their data traffic transmitted within an IPsec-protected VXLAN tunnel between the L2S VNFs.

Practical experiments and results:

During the experiments, different configuration options for the L2S VNFs were considered to evaluate the impact on performance of security mechanisms and the available CPU resources. For each configuration under test, the *iPerf* tool measured the maximum average throughput between the communication endpoints, repeating each throughput measurement 40 times to calculate the corresponding average value.

Figure 7.4 collects the results obtained from the different configurations of the L2S VNFs. In a first experiment, the security mechanisms at the L2S VNFs were disabled. This way, data traffic was exchanged unprotected between communication endpoints, using the VXLAN tunnel established between the L2S VNFs. Using 1 virtual CPU at each L2S VNF results in a maximum average throughput of 1.13 Gb/s. This value is determined by the overhead of VXLAN tunneling operations (preliminary tests indicate a baseline throughput higher than 15 Gb/s if VXLAN and IPsec procedures were disabled). In

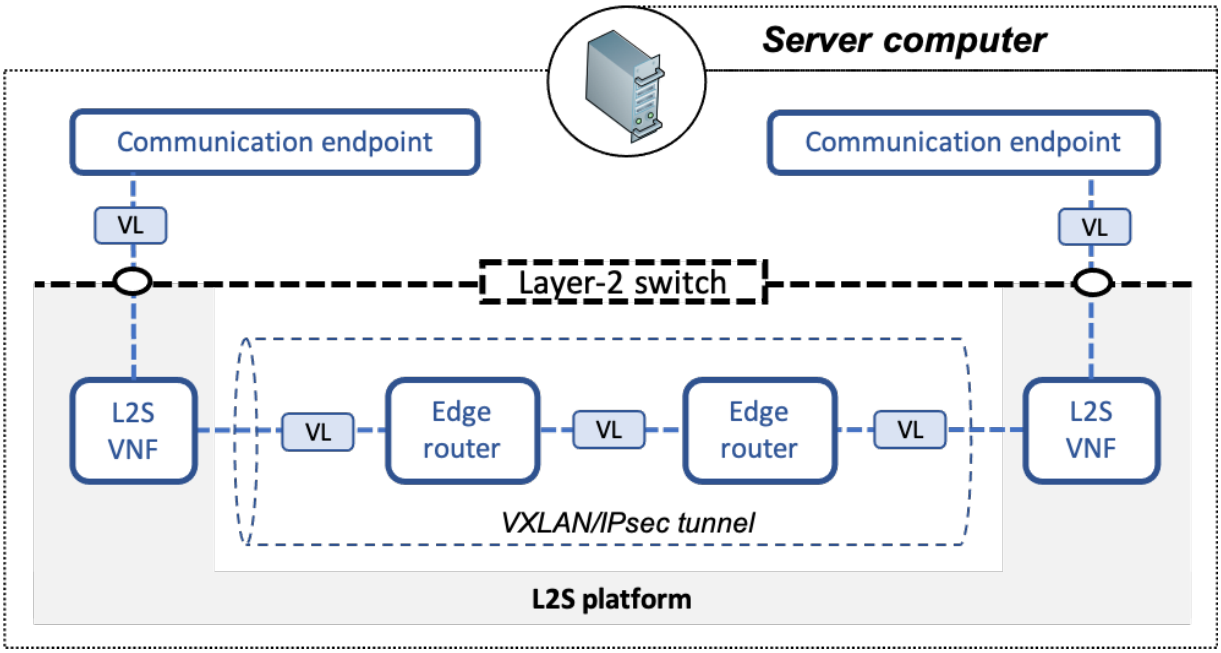


Figure 7.3. *Performance evaluation scenario.*

a second experiment, IPsec transport mode at both L2S VNFs was enabled. As can be noticed, this caused a noticeable reduction of the achievable throughput, with an average value of 556.67 Mb/s.

The use of VXLAN interfaces reduces the Maximum Transmission Unit (MTU) on L2S VLANs to 1450 bytes. This is due to VXLAN tunneling procedures, which encapsulate the MAC frames received from the communication endpoints into IP packets, appending an additional overhead to the MAC frames in the form of outer headers [89]. These packets can then be forwarded out the Ethernet interface of the L2S VNF towards the edge router, which presents a MTU of 1500 bytes (*i.e.*, the default Ethernet MTU). Hence, VXLAN interfaces support a reduced MTU value of 1450 bytes. As each VXLAN interface is connected to an access port of its corresponding L2S VNF through the *Open vSwitch* software, the MTU of all the access ports of an L2S VNF will be 1450 bytes. Consequently, this will be the MTU of any virtual link that is available for inter-site endpoint communications.

It is important to note that using IPsec causes fragmentation of data packets at L2S VNFs. As commented, a communication endpoint will generate IP packets with a size of 1450 bytes. However, after appending the IPsec protocol overhead to VXLAN encapsulated packets, these packets will exceed the MTU of the outgoing interface at the L2S VNFs, *i.e.*, 1500 bytes. The Ethernet interface of the L2S VNF will split each data packet produced by IPsec into two fragments, which will be delivered to the other VXLAN tunnel endpoint, *i.e.*, the other L2S VNF. These fragments will need to be reassembled at the receiving VXLAN tunnel endpoint. This process of fragmentation and reassembly requires computation at the L2S VNFs [159], causing a negative impact on performance. An alternative to address this well-known issue is to fragment IP packets before they are encapsulated [160], configuring an appropriate MTU value in the upstream data path, such that: *i)* fragmentation happens before IPsec

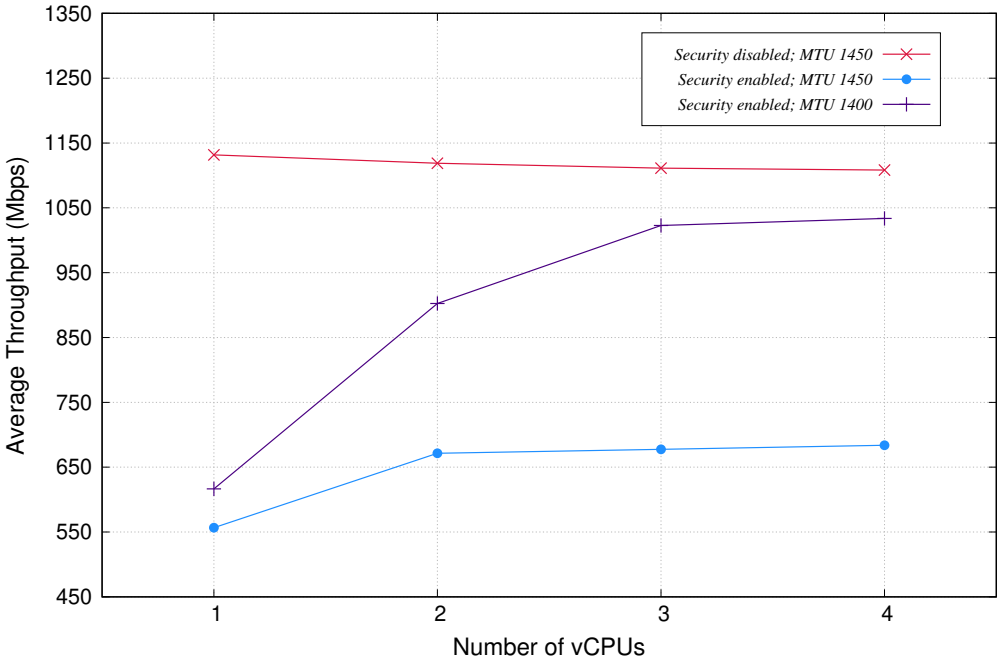


Figure 7.4. *Performance of the L2S platform implementation.*

processes; and *ii)* the VXLAN and IPsec protocol overheads do not make IP fragments exceed the MTU of the outgoing link at the L2S VNFs. This presents the additional benefit that reassembly operations are not needed at the L2S VNFs (*i.e.*, the VXLAN tunnel endpoints). Instead, they will be performed by the destination of the fragmented packets (the communication endpoints in the experiments). In this solution, this can be done by simply decreasing the MTU on the L2S VLANs to account for the IPsec protocol overhead. To this purpose, a third experiment set this MTU value to 1400 bytes. With this, the maximum average throughput results 616.5 Mb/s, *i.e.*, an increase of approximately 10.7% with respect to the case where the MTU was 1450 bytes and fragmentation and reassembly was performed at the L2S VNFs.

If the security mechanisms of IPsec are disabled, increasing the number of virtual CPUs on the L2S VNFs does not result in a positive impact on the achievable throughput. This indicates that VXLAN encapsulation and decapsulation processes cannot benefit from the parallelization offered by multiple virtual CPUs (as a matter of fact, the overhead of coordination among virtual CPUs translates into a slightly downward trend in throughput). When IPsec security mechanisms are enabled, it can observed an increase of performance with the number of virtual CPUs allocated to L2S VNFs. This is most noticeable with an MTU value of 1400 bytes on the L2S VLANs, as this value avoids the execution of fragmentation and reassembly procedures by L2S VNFs. In this case, using two CPUs increases the maximum average throughput to 902.55 Mb/s, *i.e.*, an increment of approximately 46.4% with respect to the use of a single virtual CPU. As the number of virtual CPUs increases, the achievable throughput converges to the value that can be obtained when IPsec mechanisms are disabled (with 4 virtual CPUs, the former is approximately 91.3% of the latter).

Operational and scalability considerations:

The proof-of-concept implementation of the L2S platform supports the provision of functional VLANs to multi-site NFV services. According to the aforementioned experimental results, with the allocation of 1 virtual CPU at each L2S VNF, these VLANs can operate at 100 Mb/s data rates. L2S operations can exploit multiple CPUs, increasing the achievable throughput on an L2S VLAN. This is most noticeable if fragmentation and reassembly of IP packets are not performed by L2S VNFs, which can be guaranteed with the configuration of an appropriate MTU value on the interfaces of the communication endpoints connected to the L2S VLANs. In that case, the prototype implementation can provide data rates up to 1 Gb/s. In a real scenario, the configuration of specific MTU values on endpoint interfaces can be automated using centralized management mechanisms, such as specific DHCP [91] servers provided by the NFV infrastructures, management clients based on protocols such as NETCONF [161], virtualization initialization scripts, etc. These centralized mechanisms are under the control of a VIM, which can be provisioned with the MTU values to be used on the pre-created virtual links that communication endpoints can use to connect to L2S VNFs (see Figure 7.2). This way, appropriate MTU values can be assigned to communication endpoints in a coordinated way.

The previous performance evaluation provides reference indicators for the achievable throughput on a single L2S VLAN that interconnects two sites. A decrease on the maximum average throughput per L2S VLAN could be observed as the number of active VLANs increases (*i.e.*, the VLANs actually being used). A similar situation can be expected as the average number of sites per VLAN augments. To adapt to varying traffic demands and support the scalable operation of the L2S platform, the underlying virtual nature of the platform can be exploited by increasing the allocation of resources at each L2S VNF (*i.e.*, vertical scaling), particularly virtual CPUs; and creating multiple instances of the L2S VNF at any given NFVI (*i.e.*, horizontal scaling), segregating the space of L2S VLANs into the different instances.

7.3.3. Functional validation

To validate the functionality of the L2S platform, a realistic use case was considered. In particular, an IP television (IPTV) service to be provided to a set of subscribers at their residential environments. The IPTV service has a head end, which receives the video signals from different TV content providers, and processes them to produce the compressed video formats that will be delivered to IPTV subscribers. Following a common approach for large-scale IPTV services, the head end transmits the video content of the TV channels to a set of video-serving offices. Each service office can then distribute the content to IPTV subscribers on a specific local area (*e.g.*, a municipality). The IPTV service resorts to IP multicast technologies, to support the efficient distribution of video content from the head end to IPTV subscribers.

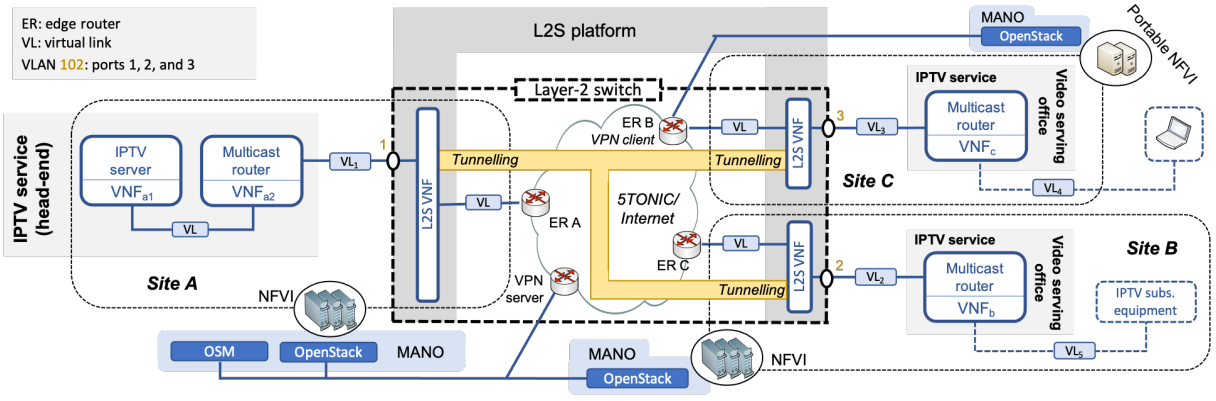


Figure 7.5. *Overview of the validation scenario.*

Description of the validation scenario:

This realistic use case was realized by deploying the validation scenario illustrated in Figure 7.5. In this scenario, the IPTV head end is represented by an IPTV server and a multicast router, both provisioned as VNFs. The IPTV server (VNF_{a1}) handles several TV channels. Each channel is assigned to a multicast host group, being its corresponding video content transmitted from the IPTV server to a specific IP multicast address. The multicast router (VNF_{a2}) receives the multicast traffic of the TV channels and distributes it towards two distant video-serving offices. Each of these offices is in turn represented by a multicast router (VNF_b and VNF_c, respectively). To support an end-to-end network path in the validation, each of these multicast routers present a link towards an IPTV subscriber equipment. In a realistic scenario, this link would typically be provided through a home gateway and an optical line termination, considering that the user has a fiber-based Internet access. Given that the L2S platform is not involved in the provision of the link towards the IPTV subscriber, the validation scenario represents one of the subscriber equipment as a virtual machine connected to its corresponding video service office through a virtual link. The other IPTV client device will be provided by a laptop, enabling the real-time visualization of TV channels.

Provision and configuration of NFV sites:

The realization of the scenario shown in Figure 7.5 requires three independent NFV sites, to host the functions of the IPTV head end and the two video-serving offices. To deploy this target scenario, the experiment utilizes the multi-site NFV ecosystem available at the 5G Telefonica Open Network Innovation Centre (5TONIC) [63]. This ecosystem includes a production-quality MANO software stack, based on the ETSI-hosted open source MANO (OSM) project, along with two independent NFVIs, each under the control of an OpenStack VIM (this NFV ecosystem is described in Chapter 3 and Chapter 4). In addition, this experiment included a third NFVI, setting up a new OpenStack controller and a set of portable mini-ITX computers as compute nodes. This NFVI was connected to the Internet through a fiber-optic access provided by a commercial Internet service provider. Following the

methodology presented in [77], this portable NFVI was integrated into the multi-site NFV ecosystem, making use of the VPN service offered by the 5TONIC laboratory. This way, all the aforementioned NFVIs were interconnected at layer-3, by means of the 5TONIC routing infrastructure and the VPN service.

With this, the realistic validation scenario presents three independent sites. The first two sites (identified as *A*, *B* in Figure 7.5) are hosted within 5TONIC, whereas a site with a portable NFVI (site *C*) is available from an external location. In addition, a virtual link in the form of a VLAN has been pre-created at each of the NFVIs (VL_1, VL_2 and VL_3) in Figure 7.5), making use of their corresponding OpenStack VIMs. These virtual links will be used to support inter-site communications among VNFs of the IPTV service. Following the previous analysis, the MTU of these virtual links is configured to 1400 bytes to avoid fragmentation and reassembly processes at the L2S VNFs. An additional virtual link has been pre-created as a VLAN in sites *B* and *C* (VL_4 and VL_5 in Figure 7.5, respectively), to support the connectivity of an IPTV subscriber equipment. In site *C*, this equipment has been represented by a laptop, which has been connected to the pre-created VLAN through an Ethernet Port of a mini-ITX computer. The IPTV subscriber equipment of site *B* has been represented as a virtual machine.

Implementation of the IPTV service functions:

The different components of the IPTV service were prototyped as VNFs implemented using Linux virtual machines (Ubuntu 16.04.5 LTS). The multicast router VNFs was built using an implementation of Protocol Independent Multicast–Sparse Mode (PIM-SM) [162], *i.e.*, the Linux *pimd* multicast routing daemon. The IPTV server VNF is based on the *VLC* media player, which can be used to stream a video content to a specific multicast host group. The laptop and the virtual machine representing the IPTV subscriber equipment also use the *VLC* tool to consume the multicast video streams. In this respect, *VLC* implements the Internet Group Management Protocol (IGMP) [163], which can be used to join and leave any multicast host group corresponding to a TV channel.

Deployment of L2S and IPTV service:

After creating the NFV descriptors for all the components shown in Figure 7.5, the last steps to realize the validation scenario were taken in two phases. In a first phase, the 5TONIC MANO platform deployed the L2S platform, resulting in the creation and configuration of an L2S VNF on each NFVI. These VNFs provided the abstraction of a layer-2 switch with an access port on the head end and on each of the video-serving offices (conforming the VLAN identified as 102 in the example).

In a second phase, the MANO platform deployed the IPTV service, following the VNF placement policies indicated in Figure 7.5. This way, the head end was created on site *A*, whereas sites *B* and *C* were used to host the functions of the video service offices. Every multicast router VNF was attached to an access port of the L2S platform, through the virtual links pre-created at the NFVIs (VL_1, VL_2 and VL_3). Finally, all the VNFs of the IPTV service were configured through *Ansible playbooks*, which can

be provided to the OSM software stack within their corresponding NFV descriptors (see Section 3.3.4). Configuration operations include setting up IP unicast addresses and network routes on every VNF, as well as the configuration and activation of the *pimd* routing daemon in the multicast routing functions.

Validation results:

After the successful deployment of the IPTV service, the multicast routing functions (VNF_{a2}, VNF_b, and VNF_c) started exchanging PIM-SM protocol messages through the L2S platform. These routing functions were effectively connected at layer-2 as expected as if they were on the same local area network, despite being placed at different NFVI sites.

The IPTV server (VNF_{a1}) was configured to start the transmission of the video corresponding to a TV channel. To emulate a realistic TV channel stream, the server transmitted the content of a high-definition video file. The video was streamed to a specific multicast host group, represented by the IP multicast address 239.0.0.10. Figure 7.7.b shows the transmission rate of the video content, measured at the multicast router of the head end (VNF_{a2}).

Approximately ten seconds after the video transmission is started, the virtual machine acting as the IPTV subscriber equipment at site *B* executed the *VLC* tool, joining the multicast host group corresponding to the TV channel (the virtual machine was already active when the IPTV service was deployed). This caused the transmission of IGMP messages by the IPTV subscriber on the virtual link that connects it with the multicast router of site *B* (VNF_b), to report its membership to the host group. In addition, the execution of the *VLC* tool resulted in the exchange of PIM-SM messages by the multicast router VNFs across the L2S platform. This created state in the multicast router VNFs, supporting the dissemination of the multicast video traffic from VNF_{a1} to VNF_b. The latter forwarded the received video traffic onto the virtual link towards the IPTV subscriber equipment of site *B*, where it was consumed.

Figure 7.7.a exhibits the throughput of IGMP traffic received and transmitted by the multicast router of site *B* (VNF_b) on the link towards the IPTV subscriber equipment, for the whole duration of the validation process. It also shows the throughput of the PIM-SM traffic observed by that multicast router. The picture reflects the periodic nature of IGMP and PIM-SM traffic, which is needed to maintain the multicast state at the different entities of the IPTV service. Figure 7.7.a shows the throughput of the video traffic delivered on the link to the IPTV subscriber equipment of site *B*, measured at the interface of VNF_b on that link.

The IPTV subscriber equipment of site *C* tuned in to the IPTV channel 30 seconds after beginning the video transmission, starting the *VLC* tool. This resulted in the corresponding execution of IGMP and PIM-SM procedures, and the consequent delivery of the solicited video on the virtual link towards the IPTV subscriber equipment of site *C*. Figure 7.7.d represents the throughput of the received video, measured at the interface of VNF_c towards the IPTV subscriber equipment.

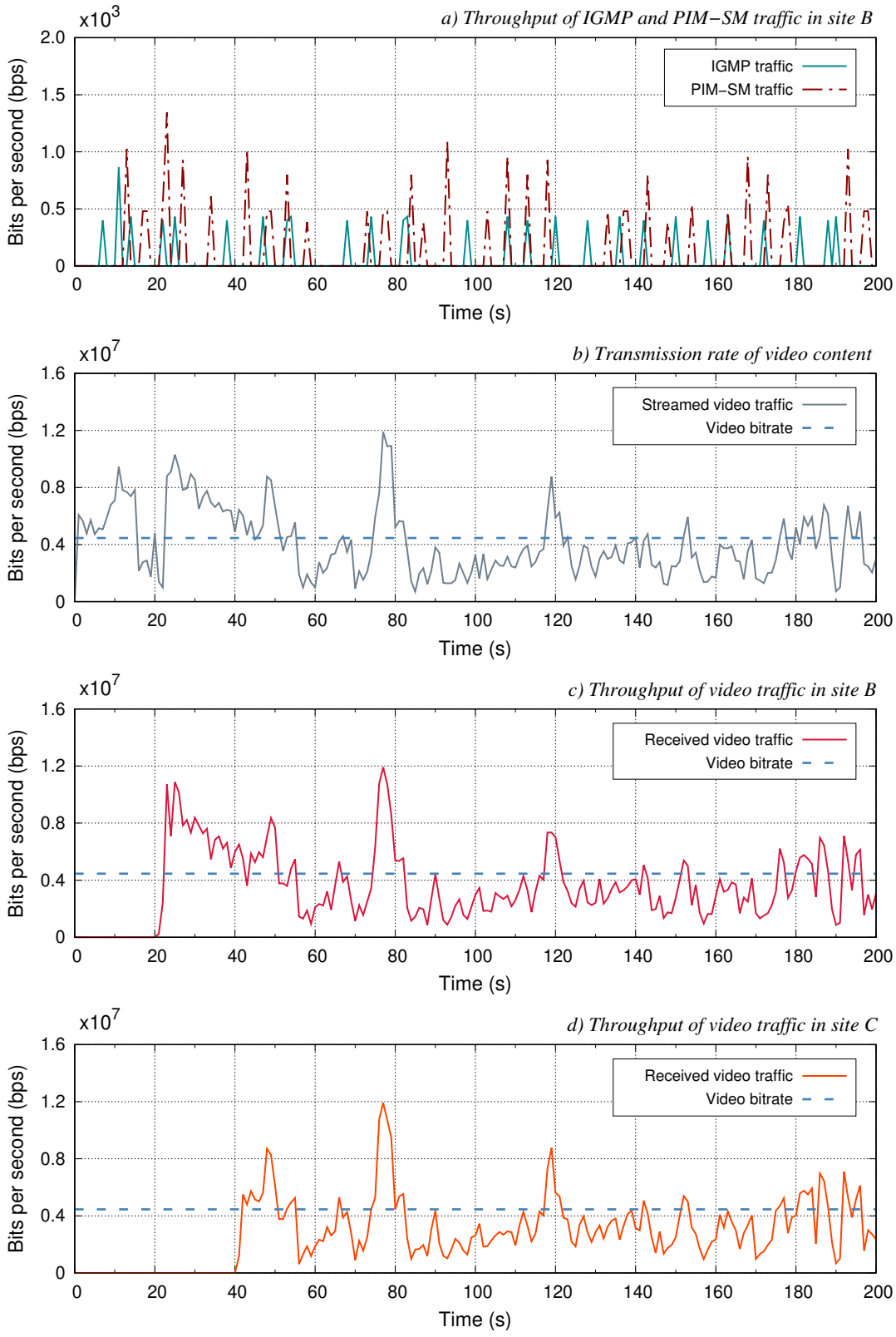


Figure 7.6. *Operation of the IPTV service.*

It is important to highlight that the platform enabled the exchange of inter-site multicast traffic among VNFs, even though multicast routing was currently disabled at the edge routers of the NFVIs (note that whereas multicast routing could be enabled within 5TONIC premises, this feature is not available over the commercial fiber-optic Internet access of the portable NFVI). The distribution of video content within the IPTV service proceeded as expected. The video was played out normally at the laptop acting as the IPTV subscriber equipment at site *C*, with no skipped or freezing video frames. The throughput of the received video streams (represented in Figures 7.7.c and 7.7.d for sites *B* and *C*, respectively) closely matched the rate at which the video was transmitted from the IPTV server (shown in Figures 7.7.b). For reference, the throughput graphs also indicate the average bitrate of the high-definition video file used in the validation. To guarantee proper timing synchronization in the throughput figures, the clocks of the three multicast router VNFs were synchronized using *chrony*, an open source implementation of the Network Time Protocol (NTP) [164]. End-to-end delays are not noticeable in the throughput measurements, given the low round-trip times that exist between sites (2.861 ms between sites *A* and *B*, and 10.589 ms between sites *A* and *C*).

7.4. Inter-domain Connectivity Orchestration Service

This section envisions an evolution of the previously presented L2S platform, supporting the automated and on-demand creation of virtual networks among sites (*i.e.*, multi-domain virtual networks). To this purpose, this section includes the design of an inter-domain connectivity orchestration service that constitutes a first approximation to support this type of multi-domain networks. Moreover, it introduces a novel architecture that embraces an Software Defined Networking (SDN) framework to support cost-effective, and reliable connectivity among heterogeneous infrastructures involved in an NFV ecosystem. Thus, promoting sharing resources in terms of inter-domain links, connecting the infrastructures involved in this sort of environments.

7.4.1. Motivation

As previously commented, a fundamental challenge to support the provision of telecommunication and vertical services in 5G consists of having appropriate mechanisms to enable the exchange of data traffic between VNFs, located in different NFV domains.

In this regard, this section introduces the inter-domain connectivity orchestration service, which allows the creation of virtual networks between different NFV domains. In this approach, the inter-domain connectivity orchestration service enables the automated provisioning and configuration of such networks on-demand, during the process of deploying a multi-domain NFV service. The VNFs of the same service that are deployed in different NFV domains and that, according to the service specification, are required to be interconnected, may be linked to an inter-domain virtual network through one of their IP interfaces. The orchestration service ensures the provision of link-level connectivity among all the VNFs connecting to the same virtual network, regardless of the domain in which each

of these VNFs is deployed. In practice, such connectivity would be provided over the underlying communication networks interconnecting the NFV domains, operated by Internet service providers that will generally be external to the NFV ecosystem operations. The proposed orchestration service allows to abstract the specific details related to the communication between NFV domains, providing the perception of a link-level network to the VNFs. Thus, VNFs of the same service, deployed in different domains, are able to operate as if they were deployed in the same local network or broadcast domain. Last but not least, the inter-domain connectivity orchestration service is intended to support the complete lifecycle management of these virtual networks, including their modification (*e.g.*, to be extended to new NFV domains), and termination.

This service ensures the opacity of the multi-domain nature of the NFV ecosystem to the users of that ecosystem, avoiding the necessity of adaptations or undesired configurations in their VNFs to operate across multiple NFV domains.

7.4.2. Design of the inter-domain connectivity orchestration service

Based on the motivation described in the previous section, the following lines focus on defining the inter-domain connectivity orchestration service design, which will allow to flexibly and dynamically orchestrate inter-site links among different NFV domains. This design is illustrated in Figure 7.7, and it encompasses different logical components that are described next.

On the one hand, the L2S platform introduced above (see Section 7.2) provides the abstraction of a layer-2 switch, which supports link-layer connectivity among VNFs deployed at remote NFV domains. This approach evolves with respect to the inter-domain connectivity orchestration service in such a way that the L2S platform can be considered as a data-plane element (hereafter, simply referred to as L2S, as illustrated in Figure 7.7) that can be deployed at the network edge of every NFV domain, and will primarily be responsible of providing a data-plane interface to support inter-domain communications. More concretely, the L2S will effectively enable the separation of inter-domain VNF communications into isolated virtual networks, as well as the configuration on-demand of these type of networks. As commented above, these isolated virtual networks will be built on top of the physical networks that interconnect the different domains, which may be provisioned by untrusted Internet service providers.

With respect to the management of the L2S instances, the deployment and configuration of these instances as VNFs on top of the infrastructures integrated within the NFV ecosystem allows its automation. In this way, a transport service provider could establish an agreement with an NFV ecosystem provider to install and deploy L2S instances at each of the sites of the ecosystem, enabling and managing the inter-domain connectivity service (as if it were any other type of service deployed within the ecosystem). This allows the L2S instances to exploit the inherent advantages of the NFV technology such as the flexibility to accommodate an L2S deployment with changing resource demands related to the traffic, for instance by incorporating additional L2S virtualized functions, or scaling them as it becomes necessary.

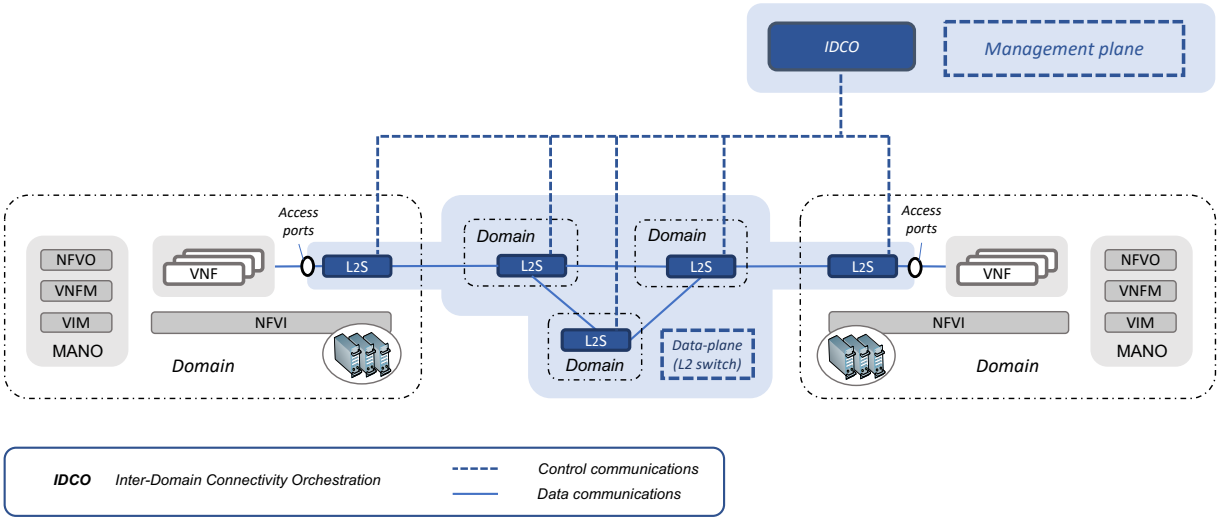


Figure 7.7. *Overview of the inter-domain connectivity orchestration service design.*

To support data-plane communications between different domains within the NFV ecosystem, the L2S instances running at different domains can be interconnected via point-to-point links, which, from an overall perspective, create an overlay network. These links can be established through the use of tunneling technologies such as Virtual eXtensible Local Area Network (VXLAN) [89]. Since the creation of these links (or in other words, tunnels) may be conditioned to underlying communications networks provisioned by untrusted Internet service providers, they can be protected (and with it, the traffic traversing those links) through the use of specific security mechanisms like IP security (IPsec) [119].

Moreover, the implementation of every L2S instance is based on a softwarized switching function. Thus, providing a programmable function, and enabling the dynamic configuration of the point-to-points links encompassed within the overlay network. This characteristic also allows, through the installation of appropriate traffic forwarding rules, to efficiently manage the communications that have as origin/destination any of the VNFs that are under the same domain as the L2S instance. This programmability is then exploited on each L2S instance by an Inter-Domain Connectivity Orchestration (IDCO) function, included within the design as an SDN framework.

This IDCO function is in charge of managing the L2S instances, by carrying out the appropriate configurations that may enable or disable the traffic traversing the access ports available at each L2S. As depicted in the figure, these access ports provide connectivity to the VNFs deployed within an NFV domain. Then, controlling the traffic traversing these access ports in each of the L2S deployed in the different domains (where the VNFs of every domain are connected), the IDCO is able to govern the overlay network, and with that, control the inter-site communications among the VNFs.

Finally, the overall design includes a management plane, supporting the control communications required from the IDCO function to realize the management of the L2S instances, as shown in Figure 7.7.

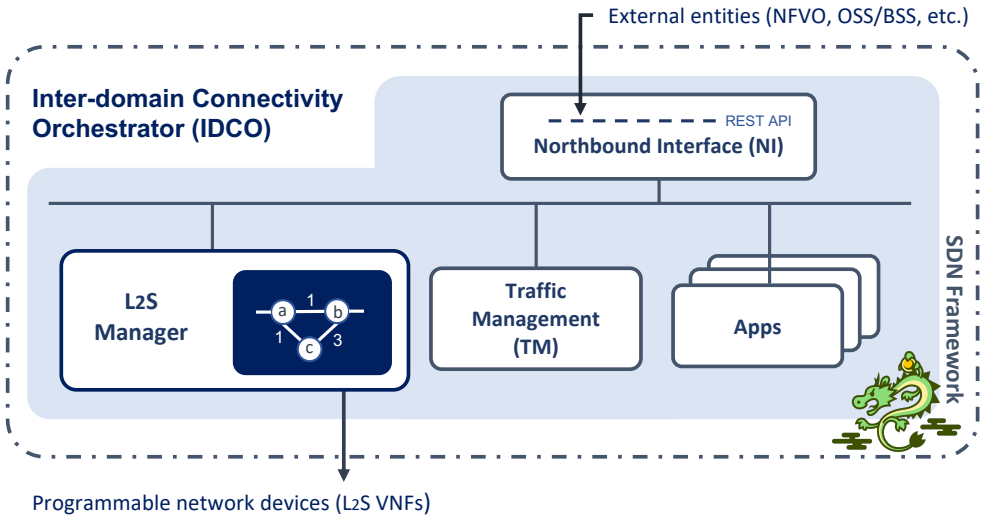


Figure 7.8. *Implementation of the Inter-Domain Connectivity Orchestrator.*

7.4.3. Implementation details of the service

This part of the chapter addresses the most relevant aspects regarding the implementation of the components building the design presented in Section 7.4.2, with particular emphasis on the Inter-domain Connectivity Orchestrator (IDCO) function.

With respect to L2S VNF, the implementation uses the prototype presented in Section 7.3. As previously commented, this prototype is based on standard Internet protocols and open source software technologies such as VXLAN, IPsec, and Open Virtual Switch (OvS).

The implementation of the IDCO function has been based on the component-based SDN framework provided by Ryu [165]. Developed in Python, and available under the open source licence Apache 2.0, Ryu has been selected as the basis for this component due to its software modularity, which allows the decomposition of the envisioned functionality into different, simpler software units, and provides a lightweight and resource-efficient solution. This expedites and simplifies the development of networks that embrace programmability-enabling elements such as L2S VNFs, and turns Ryu into an excellent candidate over other software defined framework solutions likes ONOS [75] or ODL [73], with more complex architectural implementations, and with more exhaustive computational requirements.

In particular, as illustrated in Figure 7.8, the functionality of the IDCO component has been divided into three different modules that are described next:

- The **Northbound Interface (NI)** module, whose intended purpose is to serve as an entry point to receive requests related to the inter-domain connectivity management. Thus, enabling the creation of virtual networks/links, and exposing the different operations supported at the IDCO (*e.g.*, providing the network topology, network statistics, or network status). To do this, this

module implements an application programming interface (API) based on the representational state transfer (REST) model.

- The **L2S Manager** module, which implements a southbound interface to manage the operations of the programmable network elements under its governance (*i.e.*, the L2S VNFs). To this purpose, this module leverages the OpenFlow protocol [121], widely used to directly access and manipulate the forwarding plane (also referred to as data plane) of network devices regardless of whether they are physical (switches, routers) or virtual (L2S VNF). In addition, this module is also capable of gathering information and statistics from those devices, discovering the network topology conformed by them, detecting non-expected issues (such as the failure of any of the links), and reacting to them.

  Specifically, this module represents the discovered topology by means of a graph structure, in which the nodes of the graph correspond to each network element included into the topology, and the edges of the graph represent the links between those network elements. The graph edges also enable the characterization of those links through the use of different attributes (*e.g.*, the available bandwidth, or the incurred latency). Moreover, if any change in the topology is detected through the event-driven collected information, this module would update that graph with the new obtained information.

- The **Traffic Management (TM)** module, coordinating the the paths to be followed by different data flows (*i.e.*, the flow paths). From a conceptual point of view, this module could consider different aspects, such as Service Level Agreement (SLA) policies, or the link status characterized by the L2S Manager module, to enable the application of traffic engineering principles. In this initial implementation, this module is grounded in graph theory (functionality also included in the NetworkX python package), which supports the optimal path calculation (based on the Dijkstra's algorithm) with respect to any of the attributes describing the links between the nodes of the graph.

In addition to the modules included within the IDCO component of the architecture (developed as an SDN framework), it is worth noting that, due to its modular capacity, supplementary applications/modules could be included to extend its functionality. For instance, adding a security module, capable of denying access to a VNF for security reasons, or with an additional module for monitoring traffic on a multi-domain virtual network.

Finally, and similar to the L2S, the IDCO component has been developed using open source software technologies (*i.e.*, Ryu, Python, and NetworkX), and prototyped in a virtual machine with Linux Ubuntu Server 18.04 LTS as operating system, and with the computational resources of 4GB RAM, 2 vCPUs, and 20 GB disk storage. The implementation of this framework is available under an open source license [166].

7.4.4. Experimental assessment

To validate the inter-domain connectivity orchestration service proposed in Section 7.4.2, the experimental scenario outlined in Figure 7.9 has been deployed over the same multi-site NFV ecosystem (located within the laboratory premises of 5TONIC) used for the validation of the L2S platform described in Section 7.3.3. As already mentioned, this scenario consists of three differentiated computational domains (two of them located at 5TONIC, and an external one connected through a fiber access from a commercial operator) that have served as the basis for hosting the functionalities of the proposed architecture.

Validation scenario:

These three domains comprise the NFV ecosystem upon which the functionality of the proposed service will be validated. In this context, the MANO software stack, implemented with Open Source MANO (OSM) in this scenario, coordinates the deployment of network services through the NFVIs mentioned above. Firstly, this MANO stack has been used to deploy an initial network service consisting of three L2S VNFs, each deployed in the aforementioned domains. In addition to their deployment, the MANO configures these three VNFs to perform the point-to-point links (via VXLAN tunnels) illustrated in the figure. Likewise, the scenario includes a prototype instance of the IDCO component that will be involved in registering (through the management plane provided by the 5TONIC network) the programmable network elements offered by these L2S VNFs under its governance, discovering the network topology that they constitute through their point-to-point links, and managing their forwarding plane.

Over this landscape, the MANO stack carries out the deployment of the *NS 1* service, which consists of the VNF_{1a} and VNF_{1b} virtualized functions. This service is used for functional validation tests of the solution, in which VNF_{1a} plays the role of traffic generator towards VNF_{1b} , which acts as a traffic sink. These VNFs are connected as shown in the figure: VNF_{1a} to the pre-created virtual link (VL) that provides connectivity with the port 1 of $L2S_a$, and VNF_{1b} to VL connected to the port 1 of $L2S_b$. To carry out these connections, OSM includes in the service descriptor to which pre-provisioned networks (or, in other words, to which VLs) the VNFs must be connected. After completing its instantiation, a similar procedure is performed for the deployment of the service labelled as *NS 2*, following the connection scheme depicted in the figure (*i.e.*, VNF_{2a} to the VL connected with port 2 of $L2S_a$, and VNF_{2b} to the VL connected with port 2 of $L2S_b$).

Validation process and results:

Figure 7.10 represents the set of measurements collected within the validation scenario described above with the aim of validating the operations of the novel approach presented in this work for supporting cost-effective, and dynamic link-layer inter-site communications in a multi-domain environment. To that end, the experimentation procedure included the following sequence of actions.

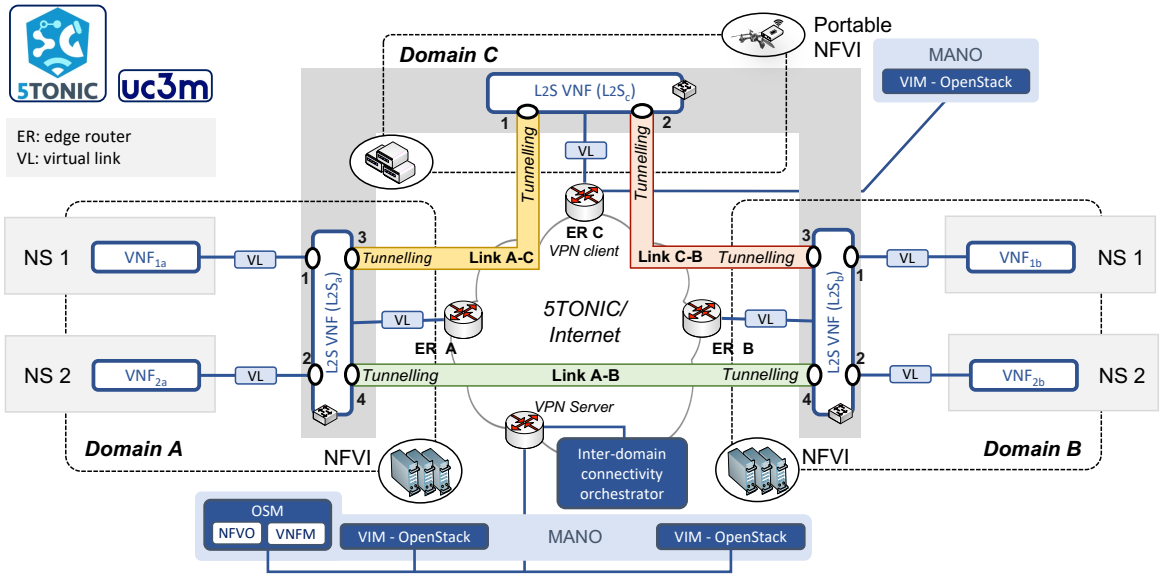


Figure 7.9. *Multi-domain scenario configured for experimentation.*

Firstly, the registration of the three L2S functionalities in the IDCO through OpenFlow. These registration events are illustrated around time instant 25 (in seconds) in Figure 7.11.c (responsible for representing the OpenFlow directives sent from the L2S Manager module to the L2S VNFs). Particularly, this allows the L2S Manager module to online discover the network topology graph comprised by the L2S VNFs.

Then, at instant 80, Figure 7.11.b shows the initial request that the IDCO receives on its NI module over the offered REST API. This event refers to the request made to the inter-domain connectivity orchestration service to obtain the status of the network topology discovered after the registration of every L2S. This allows the MANO system to obtain practical information (*e.g.*, the discovered network topology), so that it may request the creation of different flow paths that would enable the VNFs deployed on the different NFVIs to establish end-to-end layer-2 communications over a specific L2S chaining.

Approximately after 20 seconds, the MANO system executes a second API call to create a first flow path to link the VNFs of *NS 1*. For this, the MANO system posts, along with the MAC address of the instantiated VNFs (information that can be obtained at the service instantiation time by the MANO system, or even specified by itself), the ports of the L2S instances that are connected to the VNFs (information obtained in the initial API call). Thus, the TM module is able to calculate the flow path to support the above mentioned communications between VNF_{1a} and VNF_{1b} , considering the Service Level Agreement (SLA) imposed for the communications between this pair of VNFs.

In this proof of concept, the default SLA assumed at the time of creating a communications path, prioritizes those alternatives that enable the forwarding operations of a flow at a lower cost, using links that do not accommodate any other flow (whenever possible). The calculation of this cost, as commented during the design and implementation sections, may depend on different parameters.

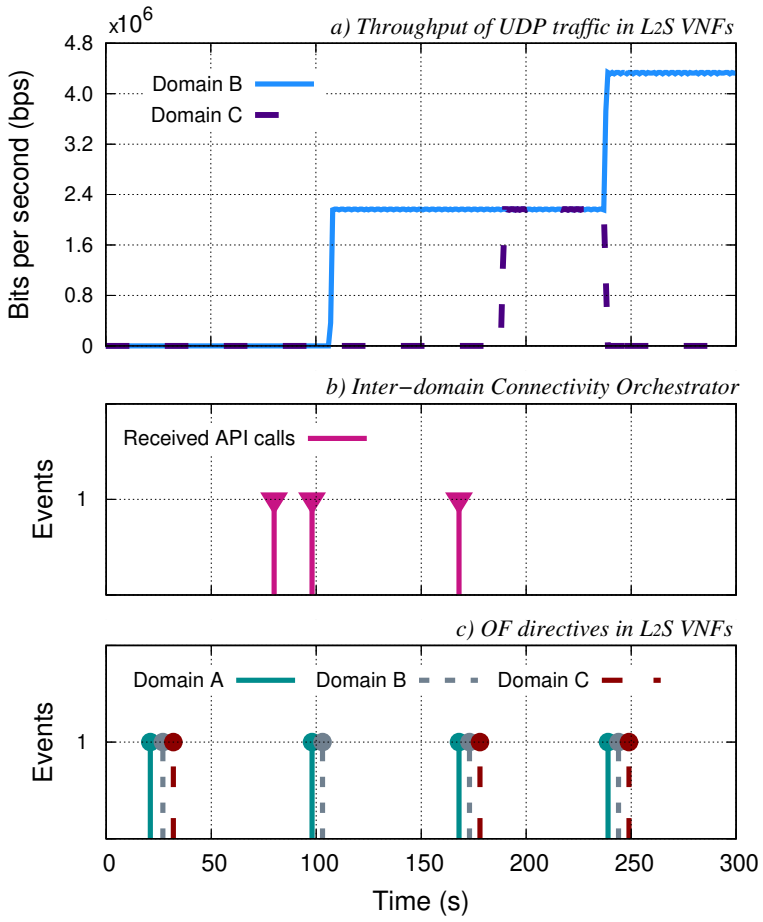


Figure 7.10. *Performance evaluation collected metrics.*

The path of L2S functions computed by the TM module for the first flow path connecting VNF_{1a} with VNF_{1b} encompasses the $L2S_a$, $L2S_b$. After this request, the L2S Manager module deals with the traffic management rules installation on the appropriate L2S VNFs (*i.e.*, the $L2S_a$ and $L2S_b$ derived from the path calculation performed by the TM module). These events are depicted at instant 98 in the plot Figure 7.11.c.

Once the IDCO reports to the MANO system that the flow path has been successfully created, VNF_{1a} triggers the transmission of a UDP traffic flow, with a 2 Mbps rate, and with VNF_{1b} as destination. To verify that this flow traverses the previously established path of L2S functions, a capture has collected the UDP traffic received through the port of the $L2S_b$ included in the calculated path (*i.e.*, port 4 of the $L2S_b$ in the figure). This is represented with the continuous blue line in Figure 7.11.a. At instant 160, the MANO system performs a new request to the IDCO in order to communicate an additional pair of VNFs: VNF_{2a} with VNF_{2b}. In this case, considering the SLA previously mentioned, the chain is established over the following sequence: $L2S_a$, $L2S_c$ and $L2S_b$. Once again, this is represented

by Figure 7.11.c. As in the preceding stage, some seconds after creating the second flow path, VNF_{2a} starts to steer UDP traffic to the VNF_{2b} instance. This is illustrated in Figure 7.11.a with the dashed purple line, representing the traffic traversing port 2 of the $L2S_c$.

In addition, Figure 7.10 illustrates how IDCO deals with unexpected conditions/events causing changes in the discovered network topology. Indeed, this is illustrated from instant 240, when, aiming at validating the ability of the IDCO component to address unexpected changes, the link between $L2S_c$ and $L2S_b$ was forced down. This generates an event that is collected by the L2S Manager, leading this module to re-calculate the stored topology, and to verify if any of the established flow paths is affected by this event. From this, the TM module re-calculates the already established flow paths, so that the SLAs may remain satisfied. Then the TM module provides this information to the L2S Manager in order to install the correspondent changes into the L2S functions. In this case, the available path is selected to accommodate both flows. As illustrated in Figure 7.11.c, the L2S Manager module addresses the management traffic rules installation at every L2S with a twofold objective: to eliminate the previously installed traffic rules associated with the affected flow path; and to install new ones in the correspondent L2S. This latter results in all traffic passing through the link connecting $L2S_a$ and $L2S_b$, which can be observed in Figure 7.11.a with the increase in the continuous blue line. This behavior demonstrates the ability of the inter-domain connectivity orchestration service to deal with unexpected situations or events.

7.5. Conclusions

This chapter has contributed this thesis with a novel solution to support secure link-layer connectivity for virtualized functions in multi-site NFV ecosystems. Thus, providing an appropriate mechanism to enable the exchange of data traffic among VNFs that are located in different NFV domains, and addressing one of the fundamental requirements identified under the context of this thesis with respect to the service provisioning in 5G networks. In this context, this chapter has presented the following contributions:

- Introduce the L2S platform, a connectivity platform that supports secure link-layer communications for multi-site NFV services. From a conceptual perspective, L2S provides the abstraction of a VLAN-capable layer-2 switch that spans multiple NFV sites. VNFs deployed on different sites can be attached to the same VLAN of the switch and be provided with link-layer connectivity with other remote VNFs. L2S protects data communications among NFV sites using existing security solutions, *i.e.*, IPsec. Moreover, the platform can be deployed on multiple NFV sites as a regular multi-site network service. Hence, the solution does not require the installation and management of additional network equipment at those sites.

- Design and implementation of a completely functional prototype of the L2S solution, demonstrating its potential of innovation using standard protocols and state-of-the-art open source technologies.

- Validation of the practical feasibility of the solution. To this purpose, this chapter has described the use of the L2S platform to support a realistic multi-site IP television service, enabling the multicast-based distribution of video content among the three NFV sites.

- Evolve the L2S platform approach, presenting the design of an inter-domain connectivity orchestration service. This service is intended to support the automated and on-demand provisioning and configuration of virtual networks between different NFV domains. Thus, ensuring the provision of link-level connectivity among all the VNFs connecting to the same virtual network, regardless of the domain in which each of these VNFs is deployed.

- Implementation of the inter-domain connectivity orchestration service design, based on an SDN framework. A remarkable aspect of this implementation is its ability, due to its modular design, to include supplementary applications/modules to enhance its functionality. Finally, this implementation has served as the basis for the results that validate the feasibility of the proposed service.

To conclude, the following chapter presents the main conclusions derived from this thesis, as well as the future research lines to be explored.

CHAPTER 8

Conclusions & Future Work

This final chapter is devoted to compiling the main lessons that have been learned with the realization of this thesis, emphasizing its main contributions, as well as to presenting the future research lines to be explored, which have emerged as a result of these lessons learned.

8.1. Conclusions

As outlined during the course of this thesis, the stream adopted for the current and fifth generation of mobile networks is to completely overturn the traditional approach in the context of mobile networks, and to involve assets that have not been considered in the provision of services. This paradigm shift includes the requirements exposed by verticals or industrial sectors in the provision of communication services, which traditionally have been exclusively dedicated to human communications. That is why 5^{th} Generation of Mobile Networks (5G) has increased the portfolio of feasible products and services, creating new market opportunities for stakeholders working in network provisioning, and in the service provider landscape.

This thesis is realized under this departure point offered by the 5G service provisioning model. In the first part, this thesis designs and experimentally evaluates the deployment of a platform to accommodate network services that involve several vertical infrastructures, distributed over different geographical locations (*i.e.*, multi-site services). To this end, this platform was based on one of the technologies identified as key enablers of 5G networks: the Network Functions Virtualization (NFV). As previously mentioned, it is important to emphasize the temporal context in which this part of the

thesis is situated (*i.e.*, in 2017), where NFV was starting to receive a great interest from the industry and research community, and there were just a few open source initiatives aiming to implement the standard. Due to this, an additional effort was needed to understand, by means of existent implementations at that time, the implications and challenges of applying the NFV standards in practical situations. Under this perspective, this part presents one of the main contributions of this thesis, since this platform (both in its design and functional profile) has served as the basis for the development of different European projects defined within the EU Horizon 2020 framework, and Spanish national projects (*e.g.*, 5GinFIRE, 5G-VINNI, 5GZORRO, 5GCity, or TRUE5G), in addition to assist in the consolidation of important aspects of the standard, such as the practical applicability of NFV in the context of verticals, or the validation of orchestration platforms to potentially support multi-site services. Moreover, this part of the thesis also includes the contribution to Open Source MANO (OSM), one of the most relevant upstream projects within the NFV ecosystem, providing an open source solution to easily perform the configuration of Virtualized Network Functions (VNFs) that are deployed with the OSM stack.

> Publications covering the design and implementation of the NFV multi-site platform, including related concepts, are [167], and [77].

> An open source implementation to perform VNFs configuration with the OSM stack is available at [85].

Based on the knowledge acquired from the first part of the thesis related to NFV, the next part aims at addressing one of the challenges faced in 5G networks with respect to the orchestration and management of services in environments and situations where there are obvious resource constraints. For instance: *(i)* in remote areas where 5G radio access network coverage is insufficient or non-existent; *(ii)* in emergency situations (e.g., natural disasters), where the network infrastructure may fail or provide deficient service; or *(iii)* in situations where there are occasional high, unexpected or predictable, service demands such as in the case of mass events.

To this end, this thesis promotes the use of Unmanned Aerial Vehicles (UAVs) due to the inherent ability of these devices to be positioned in delimited geographical areas. In this context, the thesis presents the integration of these aerial devices within an NFVs Management & Orchestration (MANO) platform, so that by coordinating the resources provided by the UAVs in terms of computation, storage and network, it may be supported the automated, cost-effective, and on-demand deployment of moderately complex services on these devices. Thus, extending the programmable substrate of 5G networks beyond the network access segments of telecommunications operators. As a major contribution, this part of the thesis includes the design and prototype implementation of the above vision, considering as a fundamental aspects the defined drivers to support the communications within this particular NFV mobile infrastructure. Additionally, this approach was validated through the deployment of a realistic telecommunications service, providing an IP telephony service to end-users in the vicinity of UAVs.

> Publications covering the design, implementation, and validation of the UAVs-based NFV infrastructure, including related concepts, are [168], [169] and [170].

Following the line of supporting adaptable and automated service deployments through and NFV system based on UAV devices, the next part of the thesis focused on exploring the potential benefits of using the UAV-based platform in the deployment of services included within the context of different vertical sectors. Firstly, this part analyzes possible synergies between NFV, UAVs, and vertical services from a practical perspective, presenting the creation of a multi-site testbed at national scale to support prototyping, and experimentation activities. Then, this is validated with the definition of a use case involving smart-farming vertical, instantiating a precision agriculture service over the UAVs on a remote site. This practical approach allowed, in the context of this thesis, to identify a problematic situation regarding the deployment times taken by the NFV platform implemented to instantiate services, related to the mechanism used by the OSM stack to carry out the configuration of the VNFs that constitute the service.

This analysis is continued then, exploring the potential of the NFV system based on UAVs to interoperate with other NFV infrastructures, and support the deployment of telecommunications and/or vertical services in resource-constrained situations. In particular, this part of the thesis considered the existing research development conducted by the Instituto de Telecomunicações of Aveiro (Portugal), which employs an NFV infrastructure based on a fleet of vehicles. As a result of the collaboration with this institute, this part of the thesis defined a comprehensive framework that can orchestrate very diverse vertical services by integrating heterogeneous infrastructures with different computing resources and capabilities. Thus, the definition of this framework involved two distinguished mobile environments and their networks: UAVs, supporting a Flying Ad-hoc Network (FANET) to accommodate on-demand the deployment of services; and vehicles, promoting a Vehicular Ad-hoc Network (VANET), to opportunistically host functionalities on real, connected vehicles. Furthermore, it includes the realization of a complex use case involving the public-safety vertical to underline the flexibility of the UAVs-based NFV system. Finally, it addresses the challenging situation related to the deployment times due to the mechanisms of VNFs configuration, providing an innovative solution of the NFV architectural component in charge of the VNF configuration (*i.e.*, the VNF Manager, or VNFM) based on the publish–subscribe model.

> Publications covering the study on the available synergies between the UAVs-based infrastructure and the vertical services, including related concepts, are [171], and [172].

> An open source implementation to provide a novel solution for the VNFM based on the based on a publish–subscribe model is available at [145].

To close this thesis, the last part is dedicated to address the challenge imposed in the multi-domain NFV environments with respect to the support of communications between VNFs that are distributed across infrastructures located in different geographical locations. These communications usually take place through untrusted network domains of Internet service providers, and are external to the NFV ecosystem. In this context, this part focused on providing a effective mechanism to

enable the data transmission between VNFs deployed over a multi-domain NFV ecosystem, which can integrate heterogeneous infrastructures with disparate resource (*e.g.*, a cloud environment, or the UAVs-based platform presented in this thesis). For this, the content included in this part analyzed the landscape of inter-domain NFV communications, and provide the design and implementation of a novel solution (referred to as L2S) to support link-layer connectivity for virtual functions in multi-site NFV ecosystems. In addition, this solution was validated with the deployment of an IPTV television based on a multi-cast communication model, which was enable through the use of the above mentioned solution. Then, this solution was evolved to integrate the Software Defined Networking (SDN) technology, so that it supports the creation of software-driven overlay networks, and exploit the possible options available to link different NFV infrastructures (not only the direct links).

> Publications covering the design, prototype implementation, and validation of the platform supporting secure link-layer communications in multi-domain NFV ecosystems, including related concepts, are [173], and [174].

> An open source implementation, based on SDN, providing an orchestration service capable of efficiently and dynamically managing the inter-domain links of a multi-domain NFV ecosystem is available at [166].

8.2. Future Work

Based on the lessons learned in the course of this thesis, two promising future lines of research to be addressed are elaborated next:

FUTURE LINE: EXPLORE THE USE OF CLOUD-NATIVE COMPUTING PLATFORMS TO SUPPORT NFV SERVICES

The first line of future research is related to exploring the integration of the cloud-native model into NFV to support the cost-effective, and efficient deployment of services. The cloud-native model advocates the design of applications based on micro-services architectures, which can be executed on containers [37], enabling a significant degree of flexibility when deploying an application since: *(i)* the micro-services can be migrated across different virtualization platforms; *(ii)* they allow to package the necessary software to be executed in an isolated manner; and *(iii)* they offer a scalable solution, which is able to adapt the offered applications to changing demands. In particular, this line could considers Kubernetes [40] since its is becoming mainstream for cloud-native adoption. According to the latest Cloud Native Computing Foundation (CNCF) 2020 survey [175], conducted on the global cloud-native community (including software and technology, financial services, and telecommunications consulting organizations), 91% of participants reported using Kubernetes in their organization, 83% of them in production state.

In this context, the integration of cloud-native technologies may introduce relevant benefits as listed next:

- The use of lightweight, portable and scalable containers, and the use of Continuous Integration and Continuous Deployment (CI/CD) methodology, offering a very appropriate substrate to undertake the development and implementation of NFV services;
- The widespread popularity of the cloud-native model and its state of adoption in both development and production environments, which brings new opportunities for developers, manufacturers and cloud service providers to enter the NFV market;
- Related to the previous point, the positive impact on innovation processes, and on the flexibility of alternatives to deploy services, enabling access to an extremely broad catalogue of virtual functions (developed under the cloud-native model, for the provision of value-added NFV services);
- And existing initiatives to migrate cloud-native technologies to edge environments, such as KubeEdge [41], OpenYurt [42] or K3s [43], focused on the use of Kubernetes. These initiatives represent a very promising option for a potentially unlimited catalogue of computing, storage and network resources for the automated deployment of the operator and vertical services of the future.

In any case, the adoption of this model poses a notorious challenge when deploying NFV services. Specifically, the connectivity service included in platforms such as Kubernetes does not have the versatility of the virtual networks included within the NFV ecosystem. The connectivity service embraced by the cloud-native model consists of ensuring that containers have a functional network interface that enables communication between micro-services, and between these and other elements outside the service (*e.g.*, devices connected to the Internet). In this regard, the CNCF provides the Container Network Interface (CNI) solution [176], a specification and a set of libraries that provide a framework for developing plugins to configure network interfaces in Linux containers. Nowadays, this framework has been adopted by different container platforms, such as Kubernetes, and there are currently multiple plugins for different platforms (Flannel [177], Calico [178] or Multus [179], to name a few examples). Therefore, the CNI reference model facilitates the creation of network interfaces in containers, and enables effective connectivity between containers through their respective network interfaces. This connectivity model is clearly appropriate for micro-services based applications, where micro-services must be able to communicate with each other.

However, while this approach to connectivity is appropriate for cloud-native applications, it is important to note that it presents limitations to deploy NFV services. In NFV, services are deployed as a set of VNFs, which are interconnected through virtual networks. Virtual networks provide the abstraction of point-to-point or multi-access links to VNFs: the virtual networks allow two or more VNFs to effectively connect to the same link-level network, and thus share the same broadcast domain in which all connected VNFs are observed as neighbours at a single IP-level hop. In addition, these virtual networks guarantee the isolation of the data traffic transmitted over them. This means that the traffic transmitted over a virtual network is not accessible to VNFs and entities outside the virtual network. Based on the above considerations, it can be observed then that the connectivity model of

cloud-native platforms is limited to support the abstraction offered by the virtual networks commonly used in the NFV ecosystem.

This opens a new research line for the future, which would be oriented towards the support of virtual networks in cloud-native management and orchestration platforms. Thus, facilitating a seamless integration of these platforms into the NFV ecosystem, and exploiting the huge potential for innovation and the flexibility of deployment options that these technologies offer for the provision of services in 5G networks and beyond.

FUTURE LINE: ANALYZE INNOVATIVE MANO SOLUTIONS TO DYNAMICALLY AND OPPORTUNISTICALLY COMPOSE AND COORDINATE NFV INFRASTRUCTURES

Finally, a future direction that may be considered is the development of innovative MANO solutions that would allow the dynamic composition and coordination of NFV infrastructures, opportunistically using devices that may exist in a particular deployment area. Examples of this equipment include mobile end-user terminals, devices that may be used in particular vertical sectors, Customer Premises Equipment (CPE), or other appliances that may be available in residential environments, smart cities, or enterprises. These devices would enable the programmable infrastructure substrate to be extended beyond the network access segments of telecommunications operators by incorporating a large and flexible catalogue of compute, storage and network resources.

Nevertheless, this vision imposes a set of fundamental challenges that will need to be addressed by VIM platforms. These include the following challenges (recently introduced at the *ETSI NFV Evolution Event* [180]):

- **Agile and dynamic incorporation of resources:** the environment under exploration is intrinsically dynamic, where different devices may appear and disappear. The VIM platform needs to be able to discover new devices and automate their incorporation into the NFV infrastructure in a reduced time-frame, considering the service level agreements established with the entities operating such devices (e.g. telecommunications operators or service providers);
- **Flexibility to integrate management services:** given the potentially heterogeneous nature of the devices under consideration, the VIM solution should be flexible to incorporate new virtual infrastructure management services. A particularly relevant aspect is the limited lifetime of battery-powered devices. The battery status of these devices should be monitored and taken into consideration as part of the virtual infrastructure management processes (*e.g.*, to select an appropriate compute node to deploy a VNF, or to anticipate the migration of a VNF).
- **Robust management of NFV infrastructure and services:** the NFV infrastructures under consideration may consist of multiple interconnected devices, building an ad-hoc network. Therefore, communications between the VIM and these devices, as well as communications between VNFs deployed in different devices, may be established through other devices in the infrastructure (following a multi-hop approach). This could be challenging, as a device becoming

inoperative (*e.g.*, as a result of a configuration applied by its owner) could indirectly disrupt orchestration operations on other devices, as well as the proper performance of services deployed over an NFV infrastructure. Likewise, within an NFV infrastructure where nodes are interconnected to form a multi-hop ad-hoc network, the failure or loss of connectivity of one or more of the node(s) can lead to transient network fragmentation scenarios. Thus, the envisioned MANO solution should support disconnected operation models, so that it can provide a certain degree of service in isolated areas of the network.

- **Efficient communication of control and data information:** the NFV infrastructure devices can have different network interfaces to support their connectivity, relying on a variety of communication technologies, both wireless (*e.g.*, Wi-Fi or cellular radio access, such as 5G or LTE) and fixed (Ethernet or optical fiber). The MANO solution should consider the different connectivity options of each device, to select the most appropriate alternative at any given time for the exchange of control information with the VIM, as well as data traffic originating and/or terminating in the VNFs running on each device.

- **Intrinsic security in control and data communications:** in the multi-device, multi-technology environment envisaged, control communications (between the VIM and NFV infrastructure devices) and data communications (between VNFs deployed on the infrastructure) may traverse numerous equipment and network domains operated by untrusted entities. This is particularly relevant in a wireless environment where devices may communicate following a multi-hop ad-hoc network configuration. In this context, security, and specifically the authentication of orchestration actions, is a fundamental aspect that requires careful attention.

Considering the above, this future line aims at materializing the outlined vision of a highly flexible, robust and secure MANO platform, which allows to manage reliable NFV services beyond the network access segments of telecommunications operators.

References

[1] Cisco, "Cisco Annual Internet Report (2018–2023)," Whitepaper, 2020, [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on May. 17, 2022).

[2] 5G-PPP, "The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services," 5G Infrastructure Public Private Partnership, 5G-PPP, Whitepaper, 2015, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf (accessed on May. 17, 2022).

[3] 5G-PPP, "Advanced 5G Network Infrastructure for the Future Internet, Public Private Partnership in Horizon 2020; Creating a Smart Ubiquitous Network for the Future Internet," 5G Infrastructure Public Private Partnership, 5G-PPP, 2020.

[4] A. A. Ateya, A. Muthanna, M. Makolkina, and A. Koucheryavy, "Study of 5G services standardization: specifications and requirements," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2018, pp. 1–6.

[5] A. Banchs, D. M. Gutierrez-Estevez, M. Fuentes, M. Boldi, and S. Provvedi, "A 5G mobile network architecture to support vertical industries," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 38–44, 2019.

[6] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges," *Computer Networks*, vol. 146, pp. 65–84, 2018.

[7] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges," *IEEE Communications surveys & tutorials*, vol. 18, pp. 236–262, 2015.

[8] 5G-PPP, "5G Innovations for New Business Opportunities," 5G Infrastructure Public Private Partnership, 5G-PPP, Whitepaper, 2017, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2017/01/5GPPP-brochure-MWC17.pdf (accessed on May. 17, 2022).

[9] ETSI, *European Telecommunications Standards Institute*, [Online]. Available: https://www.etsi.org (accessed on May. 17, 2022).

[10] ATIS, *Alliance for Telecommunications Industry Solutions*, [Online]. Available: https://www.atis.org (accessed on May. 17, 2022).

[11] J. M. H. Rábanos, L. M. Tomás, and J. M. R. Salís, *Comunicaciones móviles*. Editorial Universitaria Ramón Areces, ISBN 9788499612089, 2015.

[12] 3GPP, *45 series of Technical Specifications (TS 45)*, [Online]. Available: https://www.3gpp.org/DynaReport/45-series.htm (accessed on May. 17, 2022).

[13] 3GPP, *44 series of Technical Specifications (TS 44)*, [Online]. Available: https://www.3gpp.org/DynaReport/44-series.htm (accessed on May. 17, 2022).

[14] 3GPP, *Specification Numbering*, [Online]. Available: https://www.3gpp.org/specifications/79-specification-numbering (accessed on May. 17, 2022).

[15] 3GPP, "High Speed Downlink Packet Access (HSDPA)," 3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Overall Description, vol. Stage 2, no. Release 14, 2016.

[16] 3GPP, "High Speed Uplink Packet Access (HSUPA)," 3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Overall Description, vol. Stage 2, no. Release 14, 2016.

[17] W. Xiang, K. Zheng, and X. S. Shen, *5G mobile communications*. Springer, ISBN 978-3-319-34206-1, 2016.

[18] ITU-R M.2083-0, "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond," International Telecommunication Union, ITU-R, Series M: Mobile, radiodetermination, amateur and related satellite services, 2015.

[19] NGMN Alliance, "NGMN 5G; Next generation mobile networks," Next Generation Mobile Networks Alliance, Whitepaper, 2015.

[20] 5G-PPP, *5G empowering vertical industries*, [Online] Available: https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf (accessed on May. 17, 2022), 2016.

[21] 5G-PPP, "5G-PPP Phase-II Projects Performance KPIs," 5G Infrastructure Public Private Partnership, 5G-PPP, Annex to Programme Management Report, 2019, [Online]. Available: https://bscw.5g-ppp.eu/pub/bscw.cgi/312793 (accessed on May. 17, 2022).

[22] 3GPP, *System architecture milestone of 5G Phase 1 is achieved, 2017*, [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1930-sys_architecture (accessed on May. 17, 2022).

[23] 3GPP TR 21.915, "Release 15 Description; Summary of Rel-15 Work Items," 3rd Generation Partnership Project, 3GPP, Technical Specification Group Services and System Aspects, 2018.

[24] ONF TR-521, "Software Defined Networking (SDN) Architecture; Issue 1.1," Open Networking Foundation, ONF, 2016.

[25] M. Chiosi *et al.*, "Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges & Call for Action," European Telecommunications Standards Institute (ETSI), Whitepaper, 2017.

[26] ETSI GS NFV 002 V1.2.1, "Network Functions Virtualization (NFV); Architectural Framework," European Telecommunications Standards Institute, ETSI, 2014.

[27] ETSI Open Source MANO (OSM), *An open source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV*, [Online]. Available: https://osm.etsi.org (accessed on May. 17, 2022).

[28] Cloudify, *Cloudify Orchestration Platform - Multi Cloud, Cloud Native & Edge*, [Online]. Available: https://cloudify.co (accessed on May. 17, 2022).

[29] Open Baton, *An open source reference implementation of the ETSI Network Function Virtualization MANO specification*, [Online]. Available: http://openbaton.org (accessed on May. 17, 2022).

[30] The Linux Foundation Projects, *Open Network Automation Platform (ONAP)*, [Online]. Available: https://www.onap.org (accessed on May. 17, 2022).

[31] ETSI OpenVIM, *A light implementation of an NFV VIM contributed to the OSM project*, [Online]. Available: https://osm.etsi.org (accessed on May. 17, 2022).

[32] OpenStack, *Open source software for creating private and public clouds*, [Online]. Available: https://www.openstack.org (accessed on May. 17, 2022).

[33] VMware, *VMware Cloud Director*, [Online]. Available: https://www.vmware.com/products/cloud-director.html (accessed on May. 17, 2022).

[34] Amazon Web Services, *Amazon Elastic Compute Cloud (Amazon EC2)*, [Online]. Available: https://www.vmware.com/products/cloud-director.html (accessed on May. 17, 2022).

[35] Microsoft Azure, *A cloud computing service operated by Microsoft*, [Online]. Available: https://azure.microsoft.com (accessed on May. 17, 2022).

[36] Juju, *operate big software at scale on any cloud*, [Online]. Available: https://juju.is (accessed on May. 17, 2022).

[37] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. lightweight virtualization: a performance comparison," in *2015 IEEE International Conference on Cloud Engineering*, IEEE, 2015, pp. 386–393.

[38] Canonical, *Infrastructure for container projects*, [Online]. Available: https://linuxcontainers.org (accessed on May. 17, 2022).

[39] Docker, *Empowering App Development for Developers*, [Online]. Available: https://www.docker.com (accessed on May. 17, 2022).

[40] Linux Foundation, *Production-Grade Container Orchestration*, [Online]. Available: https://kubernetes.io (accessed on May. 17, 2022).

[41] Linux Foundation, *KubeEdge: A Kubernetes Native Edge Computing Framework*, [Online]. Available: https://kubeedge.io (accessed on May. 17, 2022).

[42] Linux Foundation, *OpenYurt: An open platform that extends upstream Kubernetes to Edge*, [Online]. Available: https://openyurt.io (accessed on May. 17, 2022).

[43] Linux Foundation, *K3s: Lightweight Kubernetes*, [Online]. Available: https://k3s.io (accessed on May. 17, 2022).

[44] Eclipse Foundation, *Eclipse fog05: The End-to-End Compute, Storage and Networking Virtualization solution*, [Online]. Available: https://fog05.io (accessed on May. 17, 2022).

[45] G. Rigazzi, J.-P. Kainulainen, C. Turyagyenda, A. Mourad, and J. Ahn, "An edge and fog computing platform for effective deployment of 360 video applications," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, IEEE, 2019, pp. 1–6.

[46] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[47] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, 2018.

[48] V. Sharma, K. Srinivasan, H.-C. Chao, K.-L. Hua, and W.-H. Cheng, "Intelligent deployment of UAVs in 5G heterogeneous communication environment for improved coverage," *Journal of Network and Computer Applications*, vol. 85, pp. 94–105, 2017.

[49] Y. Huo, X. Dong, T. Lu, W. Xu, and M. Yuen, "Distributed and multilayer UAV networks for next-generation wireless communication and power transfer: A feasibility study," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7103–7115, 2019.

[50] L. Santesteban, S. Di Gennaro, A Herrero-Langreo, C Miranda, J. Royo, and A Matese, "High-resolution UAV-based thermal imaging to estimate the instantaneous and seasonal variability of plant water status within a vineyard," *Agricultural Water Management*, vol. 183, pp. 49–59, 2017.

[51] D. Albani, T. Manoni, D. Nardi, and V. Trianni, "Dynamic UAV swarm deployment for non-uniform coverage," in *Proceedings of the 17th international conference on autonomous agents and multiagent systems*, 2018, pp. 523–531.

[52] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "UAVs for smart cities: Opportunities and challenges," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2014, pp. 267–273.

[53] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[54] M. Khosravi, S. Enayati, H. Saeedi, and H. Pishro-Nik, "Multi-purpose drones for coverage and transport applications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3974–3987, 2021.

[55] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *2015 IEEE wireless communications and networking conference workshops (WCNCW)*, IEEE, 2015, pp. 329–334.

[56] W. Zafar and B. M. Khan, "Flying ad-hoc networks: Technological and social implications," *IEEE Technology and Society Magazine*, vol. 35, no. 2, pp. 67–74, 2016.

[57] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[58] X. Zhang, H. Wang, and H. Zhao, "An SDN framework for UAV backbone network towards knowledge centric networking," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2018, pp. 456–461.

[59] L. Bertizzolo *et al.*, "SwarmControl: An automated distributed control framework for self-optimizing drone networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, IEEE, 2020, pp. 1768–1777.

[60] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma, and Z. Liu, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3664–3674, 2020.

[61] C. Rametta and G. Schembra, "Designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Future Internet*, vol. 9, no. 1, p. 8, 2017.

[62] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, and S. Zhang, "Communication and networking of UAV-based systems: Classification and associated architectures," *Journal of Network and Computer Applications*, vol. 84, pp. 93–108, 2017.

[63] 5TONIC, *An Open Research and Innovation Laboratory focusing on 5G Technologies*, [Online]. Available: https://www.5tonic.org (accessed on May. 17, 2022).

[64] European H2020 5GINFIRE Project, *Evolving FIRE into a 5G–Oriented Experimental Playground for Vertical Industries*, [Online]. Available: https://5ginfire.eu (accessed on May. 17, 2022).

[65] European H2020 SOFTFIRE Project, *Software Defined Networks and Network Function Virtualization Testbed within FIRE+*, [Online]. Available: https://www.softfire.eu (accessed on May. 17, 2022).

[66] European H2020 FUTEBOL Project, *Federated Union of Telecommunications Research Facilities for an EU-Brasil Open Laboratory*, [Online]. Available: http://www.ict-futebol.org.br (accessed on May. 17, 2022).

[67] European H2020 ORCA Project, *Orchestration and Reconfiguration Control Architecture*, [Online]. Available: https://www.orca-project.eu (accessed on May. 17, 2022).

[68] European H2020 5G–VINNI Project, *5G Verticals Innovation Infrastructure*, [Online]. Available: https://www.5g-vinni.eu (accessed on May. 17, 2022).

[69] European H2020 5G–EVE Project, *5G European Validation platform for Extensive trials*, [Online]. Available: https://www.5g-eve.eu (accessed on May. 17, 2022).

[70] European Commission, *European Digital Innovation Hubs landscape*, [Online]. Available: https://s3platform.jrc.ec.europa.eu/dihs-per-country (accessed on May. 17, 2022).

[71] R. Verdone and A. Manzalini, "5G Experimental Facilities in Europe," NetWorld 2020 European Technology Platform, Whitepaper, Version 11.0, 2016.

[72] A. Israel *et al.*, "OSM Release THREE, A Technical Overview," ETSI OSM Community, Whitepaper, 2017.

[73] The Linux Foundation, *OpenDaylight (ODL), modular open platform for customizing and automating networks*, [Online]. Available: https://www.opendaylight.org (accessed on May. 17, 2022).

[74] Project Floodlight, *Floodlight OpenFlow Controller (OSS), community-developed, open source, Java OpenFlow controller*, [Online]. Available: https://github.com/floodlight/floodlight (accessed on May. 17, 2022).

[75] Open Networking Foundation, *Open Network Operating System (ONOS), open source SDN controller for building next-generation SDN/NFV solutions*, [Online]. Available: https://opennetworking.org/onos (accessed on May. 17, 2022).

[76] P. Neves *et al.*, "Future mode of operations for 5G – The SELFNET approach enabled by SDN/NFV," *Computer Standards & Interfaces*, vol. 54, pp. 229–246, 2017.

[77] B. Nogales *et al.*, "Integration of 5G Experimentation Infrastructures into a Multi-Site NFV Ecosystem," *JoVE (Journal of Visualized Experiments)*, vol. e61946, 2021. DOI: 10.3791/61946.

[78] iptables, *Command line utility for configuring Linux kernel firewall*, [Online]. Available: https://wiki.archlinux.org/title/iptables (accessed on May. 17, 2022).

[79] Juniper Networks, *M7i Router, a multi-service edge router*, [Online]. Available: https://www.juniper.net/documentation/en_US/release-independent/junos/information-products/pathway-pages/m-series/m7i/index.html (accessed on May. 17, 2022).

[80] OSM Wiki, *Creating your own VNF charm (Release THREE)*, [Online]. Available: https://osm.etsi.org/wikipub/index.php/Creating_your_own_VNF_charm_(Release_THREE) (accessed on May. 17, 2022).

[81] Juju, *Charm Layers Index*, [Online]. Available: https://github.com/juju/layer-index (accessed on May. 17, 2022).

[82] ETSI OSM, *Juju charm layer vnfproxy*, [Online]. Available: https://github.com/charmed-osm/vnfproxy (accessed on May. 17, 2022).

[83] OSM Wiki, *VNF onboarding guidelines*, [Online]. Available: https://osm.etsi.org/docs/vnf-onboarding-guidelines/00-introduction.html (accessed on May. 17, 2022).

[84] J. Geerling, *Ansible for DevOps: Server and configuration management for humans*. Leanpub, ISBN 978-0-9863934-0-2, 2015.

[85] B. Nogales and I. Vidal, *Juju charm layer ansible-charm*, [Online]. Available: https://github.com/5GinFIRE/mano/tree/master/charms/ansible-charm (accessed on May. 17, 2022).

[86] OSM Wiki, *OSM Information Model (IM)*, [Online]. Available: https://osm.etsi.org/wikipub/index.php/OSM_Information_Model (accessed on May. 17, 2022).

[87] OSM Wiki, *Example of VNF Charms*, [Online]. Available: https://osm.etsi.org/wikipub/index.php/Example_VNF_Charms (accessed on May. 17, 2022).

[88] Advanced 5G Network Infrastructure for the Future Internet, "Creating a Smart Ubiquitous Network for the Future Internet," Public Private Partnership in Horizon 2020, Whitepaper, 2018, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/Advanced-5G-Network-Infrastructure-PPP-in-H2020_Final_November-2013.pdf (accessed on May. 17, 2022).

[89] M. Mahalingam *et al.*, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks.," *Internet Engineering Task Force, RFC 7348*, 2014.

[90] R. Fielding *et al.*, "Hypertext Transfer Protocol: HTTP/1.1," *Internet Engineering Task Force, RFC 2616*, 1999.

[91] R. Droms, "Dynamic Host Configuration Protocol," *Internet Engineering Task Force, RFC 2131*, 1997.

[92] 5GinFIRE GitHub repository, *Open-source NS and VNF descriptors*, [Online]. Available: https://github.com/5GinFIRE/mano/tree/master/descriptor-packages (accessed on May. 17, 2022).

[93] GStreamer, *Open source multimedia framework*, [Online]. Available: https://gstreamer.freedesktop.org (accessed on May. 17, 2022).

[94] IETF, "Transmission Control Protocol," *Internet Engineering Task Force, RFC 793*, 1981.

[95] Spanish 5GCity Project, *Adaptive Management of 5G Services to Support Critical Events in Cities*, Spanish national project funded by the Spanish Ministry of Economy and Competitiveness.

[96] European 5GRANGE Project, *Remote area Access Network for the 5th GEneration*, [Online]. Available: http://5g-range.eu (accessed on May. 17, 2022).

[97] European H2020 FISHY Project, *A coordinated framework for cyber resilient supply chain systems over complex ICT infrastructures*, [Online]. Available: https://fishy-project.eu (accessed on May. 17, 2022).

[98] European H2020 5GZORRO Project, *Zero-tOuch secuRity and tRust for ubiquitous cOmputing and connectivity in 5G networks*, [Online]. Available: https://www.5gzorro.eu (accessed on May. 17, 2022).

[99] Spanish TRUE5G Project, *Towards zeRo toUch nEtwork and services for beyond 5G*, Spanish national project funded by the Spanish Ministry of Science and Innovation.

[100] I. Vidal, F. Valera, M. A. Díaz, and M. Bagnulo, "Design and practical deployment of a network-centric remotely piloted aircraft system," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 22–29, 2014.

[101] K. R. Branco, J. M. Pelizzoni, L. O. Neris, O. Trindade, F. S. Osório, and D. F. Wolf, "Tiriba – a new approach of UAV based on model driven development and multiprocessors," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 1–4.

[102] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *Australasian Joint Conference on Artificial Intelligence*, Springer, 2007, pp. 1–13.

[103] S. Waharte, N. Trigoni, and S. Julier, "Coordinated search with a swarm of UAVs," in *2009 6th ieee annual communications society conference on sensor, mesh and ad hoc communications and networks workshops*, IEEE, 2009, pp. 1–3.

[104] Y. Jin, A. A. Minai, and M. M. Polycarpou, "Cooperative real-time search and task allocation in uav teams," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, IEEE, vol. 1, 2003, pp. 7–12.

[105] L. Merino, F. Caballero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[106] M. Quaritsch *et al.*, "Collaborative microdrones: Applications and research challenges," in *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, 2008, pp. 1–7.

[107] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked UAVs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[108] A. Sivakumar and C. K.-Y. Tan, "UAV swarm coordination using cooperative control for establishing a wireless communications backbone," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 3-Volume 3*, 2010, pp. 1157–1164.

[109] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *Internet Engineering Task Force, RFC 3561*, 2003.

[110] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *Internet Engineering Task Force, RFC 3561*, 2003.

[111] Parrot, *Bebop 2*, [Online]. Available: https://www.parrot.com/assets/s3fs-public/2021-09/bebop-2_user-guide_uk.pdf (accessed on May. 17, 2022).

[112] V. Sanchez-Aguero, B. Nogales, F. Valera, and I. Vidal, "Investigating the deployability of VoIP services over wireless interconnected micro aerial vehicles," *Internet Technology Letters*, vol. 1, no. 5, e40, 2018.

[113] P. V. Mockapetris, "Domain names-concepts and facilities," *Internet Engineering Task Force, RFC 1034*, 1987.

[114] Kamailio, *The Open Source SIP Server*, [Online]. Available: https://www.kamailio.org (accessed on May. 17, 2022).

[115] J. Rosenberg *et al.*, "SIP: session initiation protocol," *Internet Engineering Task Force, RFC 3261*, 2002.

[116] 3GPP TS 23.501, "System Architecture for the 5G System; Stage 2, version 16.3.0," 3rd Generation Partnership Project, 3GPP, Technical Specification Group Services and System Aspects, 2019.

[117] 3GPP TS 23.502, "Procedures for the 5G System; Stage 2 version 16.2.0," 3rd Generation Partnership Project, 3GPP, Technical Specification Group Services and System Aspects, 2019.

[118] D. Farinacci, T Li, S Hanks, D Meyer, and P Traina, "Generic Routing Encapsulation (GRE)," *Internet Engineering Task Force, RFC 2784*, 2000.

[119] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap," *Internet Engineering Task Force, RFC 6071*, 2011.

[120] ITU-T Recommendation G.114, "One-way transmission time," International Telecommunication Union, ITU-T, Series G: Transmission Systems and Media, Digital Systems and Networks, 2003.

[121] Open Networking Foundation (ONF), *OpenFlow Switch Specification v1.0-v1.5*, [Online]. Available: https://opennetworking.org/software-defined-standards/specifications/ (accessed on May. 17, 2022).

[122] I. Vidal *et al.*, "A multi-site NFV testbed for experimentation with SUAV-based 5G vertical services," *IEEE access*, vol. 8, pp. 111 522–111 535, 2020. DOI: 10.1109/ACCESS.2020.3001985.

[123] RedIRIS, *The Spanish academic and research network for advanced communication services provisioning*, [Online]. Available: https://www.rediris.es (accessed on May. 17, 2022).

[124] M. P. Christiansen, M. S. Laursen, R. N. Jørgensen, S. Skovsen, and R. Gislum, "Designing and testing a UAV mapping system for agricultural field surveying," *Sensors*, vol. 17, no. 12, p. 2703, 2017.

[125] B. H. Y. Alsalam, K. Morton, D. Campbell, and F. Gonzalez, "Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture," in *2017 IEEE Aerospace Conference*, IEEE, 2017, pp. 1–12.

[126] A. Barrientos *et al.*, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[127] OASIS Standard, "MQTT Version 5.0," Organization for the Advancement of Structured Information Standards, OASIS, 2019.

[128] Trafic, *A traffic mix generator based on Iperf3*, [Online]. Available: https://github.com/mami-project/trafic (accessed on May. 17, 2022).

[129] H. Schulzrinne, S. Casner, R Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *Internet Engineering Task Force, RFC 3550*, 2003.

[130] Mainflux, *Open source IoT Platform Edge computing and Consulting services*, [Online]. Available: https://www.mainflux.com (accessed on May. 17, 2022).

[131] 5GRANGE, *open source repository of NFV packages and descriptors*, [Online]. Available: https://vm-images.netcom.it.uc3m.es/5GRANGE/ (accessed on May. 17, 2022).

[132] 5G-PPP Architecture Working Group, "View on 5G Architecture (Version 2.0)," 5G Infrastructure Public Private Partnership, 5G-PPP, Whitepaper, 2017, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017_For-Public-Consultation.pdf (accessed on May. 17, 2022).

[133] A. De la Oliva *et al.*, "5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 78–84, 2018.

[134] M. Silva, M. Luis, and S. Sargento, "Edge Virtualization in Multihomed Vehicular Networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2020, pp. 1–6.

[135] M. Luís *et al.*, "Exploring Cloud Virtualization over Vehicular Networks with Mobility Support," in *Connected and Autonomous Vehicles in Smart Cities*, CRC Press, 2020, pp. 223–258.

[136] A. P. Silva *et al.*, "5GinFIRE: An end-to-end open5G vertical network function ecosystem," *Ad Hoc Networks*, vol. 93, p. 101 895, 2019.

[137] B. Nogales *et al.*, "Using Aerial and Vehicular NFV Infrastructures to Agilely Create Vertical Services," *Sensors*, vol. 21, no. 4, p. 1342, 2021. DOI: 10.3390/s21041342.

[138] 5G-PPP, "5G Automotive Vision," 5G Infrastructure Public Private Partnership, 5G-PPP, Whitepaper, 2015, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf (accessed on May. 17, 2022).

[139] Ayuntamiento de Madrid, *SAMUR - Proteccion Civil, Atencion Sanitaria de Urgencias*, [Online]. Available: https://t.ly/2ZRT (accessed on May. 17, 2022).

[140] V.-N. Pham, V. Nguyen, T. D. Nguyen, and E.-N. Huh, "Efficient Edge-Cloud Publish/Subscribe Broker Overlay Networks to Support Latency-Sensitive Wide-Scale IoT Applications," *Symmetry*, vol. 12, no. 1, p. 3, 2020.

[141] P. Lv, L. Wang, H. Zhu, W. Deng, and L. Gu, "An IoT-oriented privacy-preserving publish/subscribe model over blockchains," *IEEE Access*, vol. 7, pp. 41 309–41 314, 2019.

[142] L. Duan, C.-A. Sun, Y. Zhang, W. Ni, and J. Chen, "A comprehensive security framework for publish/subscribe-based IoT services communication," *IEEE Access*, vol. 7, pp. 25 989–26 001, 2019.

[143] I. M. Wirawan, I. D. Wahyono, G. Idfi, and G. R. Kusumo, "Iot communication system using publish-subscribe," in *2018 International Seminar on Application for Technology of Information and Communication*, IEEE, 2018, pp. 61–65.

[144] Apache Kafka, *An open-source distributed event streaming platform*, [Online]. Available: https://kafka.apache.org (accessed on May. 17, 2022).

[145] B. Nogales and I. Vidal, *Publish–Subscribe Configuration Function repository*, [Online]. Available: https://github.com/Borjand/pscf-solution (accessed on May. 17, 2022).

[146] iPerf, *The Ultimate Speed Test Tool for TCP, UDP and SCTP*, [Online]. Available: https://iperf.fr (accessed on May. 17, 2022).

[147] Ping, *Network Utility for Checking Connection Status*. [Online]. Available: https://packages.debian.org/buster/iputils-ping (accessed on May. 17, 2022).

[148] 3GPP TS 22.125, "Unmanned Aerial System (UAS) support in 3GPP," 3rd Generation Partnership Project, 3GPP, Technical Specification Group Services and System Aspects, 2020.

[149] J. R. Martinez, D. R. Lopez, C. Tranoris, I. Vidal, and A. Gavras, "Experimentation over Distributed 5G NFV-Based Environments," 5GinFIRE, Whitepaper, 2019, [Online]. Available: https://doi.org/10.5281/zenodo.3568720 (accessed on May. 17, 2022).

[150] K. Mahmood *et al.*, "Design of 5G end-to-end facility for performance evaluation and use case trials," in *2019 IEEE 2nd 5G World Forum (5GWF)*, IEEE, 2019, pp. 341–346.

[151] M. Gupta *et al.*, "The 5G EVE end-to-end 5G facility for extensive trials," in *2019 IEEE international conference on communications workshops (ICC workshops)*, IEEE, 2019, pp. 1–5.

[152] I. Vidal, B. Nogales, D. Lopez, J. Rodríguez, F. Valera, and A. Azcorra, "A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services," *Electronics*, vol. 10, no. 15, p. 1868, 2021. DOI: 10.3390/electronics10151868.

[153] IEEE, "IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security," *IEEE Std 802.1AE-2018 (Revision of IEEE Std 802.1AE-2006)*, pp. 1–239, 2018. DOI: 10.1109/IEEESTD.2018.8585421.

[154] L. Andersson *et al.*, "Framework for Layer 2 Virtual Private Networks (L2VPNs)," *Internet Engineering Task Force, RFC 4664*, 2006.

[155] Y. El Mghazli, T. D. Nadeau, M. Boucadair, K. H. Chan, and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management," *Internet Engineering Task Force, RFC 4176*, 2005.

[156] ETSI GS NFV 006 V2.1.1, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Architectural Framework Specification," European Telecommunications Standards Institute, ETSI, 2021.

[157] Open vSwitch (OvS), *An open-source, programmable, production-quality virtual switch*, [Online]. Available: https://www.openvswitch.org (accessed on May. 17, 2022).

[158] strongSwan, *An open-source IPsec implementation for Linux*, [Online]. Available: https://www.strongswan.org (accessed on May. 17, 2022).

[159] P. Savola, "MTU and Fragmentation Issues with In-the-Network Tunneling," *Internet Engineering Task Force, RFC 4459*, 2006.

[160] Cisco, "Cisco VPN Services Port Adapter Configuration Guide," in. Cisco, 2008, ch. Configuring IPSec VPN Fragmentation and MTU, [Online]. Available: https://www.cisco.com/c/en/us/td/docs/interfaces_modules/services_modules/vspa/configuration/guide/ivmsw_book/ivmvpnb.html (accessed on May. 17, 2022).

[161] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," *Internet Engineering Task Force, RFC 6241*, 2011.

[162] B. Fenner, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification)," *Internet Engineering Task Force, RFC 7761*, 2016.

[163] B. Cain, S. Deering, B. Fenner, and I. Kouvelas, "Internet Group Management Protocol (IGMP), Version 3," *Internet Engineering Task Force, RFC 3376*, 2002.

[164] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," *Internet Engineering Task Force, RFC 5905*, 2010.

[165] Ryu, *component-based software defined networking framework*, [Online]. Available: https://ryu-sdn.org (accessed on May. 17, 2022).

[166] B. Nogales, *Inter-domain Connectivity Orchestrator (IDCO) software-defined framework repository*, [Online]. Available: https://github.com/Networks-it-uc3m/software-driven-l2-communications (accessed on May. 17, 2022).

[167] B. Nogales, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso, and A. Azcorra, "Design and Deployment of an Open Management and Orchestration Platform for Multi-site NFV Experimentation," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 20–27, 2019. DOI: 10.1109/MCOM.2018.1800084.

[168] B. Nogales, V. Sanchez-Aguero, I. Vidal, and F. Valera, "Adaptable and Automated Small UAV Deployments via Virtualization," *Sensors*, vol. 18, no. 12, p. 4116, 2018. DOI: 10.3390/s18124116.

[169] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A NFV system to support configurable and automated multi-UAV service deployments," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications - DroNet'18*, Munich, Germany, 10–15 June: MobiSys 2018, Association for Computing Machinery, 2018, pp. 39–44. DOI: 10.1145/3213526.3213534.

[170] B. Nogales, I. Vidal, V. Sanchez-Aguero, F. Valera, L. F. Gonzalez, and A. Azcorra, "Automated deployment of an Internet protocol telephony service on unmanned aerial vehicles using network functions virtualization," *JoVE (Journal of Visualized Experiments)*, no. 153, e60425, 2019. DOI: 10.3791/60425.

[171] I. Vidal *et al.*, "A multi-site NFV testbed for experimentation with SUAV-based 5G vertical services," *IEEE access*, vol. 8, pp. 111 522–111 535, 2020. DOI: 10.1109/ACCESS.2020.3001985.

[172] B. Nogales *et al.*, "Using Aerial and Vehicular NFV Infrastructures to Agilely Create Vertical Services," *Sensors*, vol. 21, no. 4, p. 1342, 2021. DOI: 10.3390/s21041342.

[173] I. Vidal, B. Nogales, D. Lopez, J. Rodríguez, F. Valera, and A. Azcorra, "A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services," *Electronics*, vol. 10, no. 15, p. 1868, 2021. DOI: 10.3390/electronics10151868.

[174] B. Nogales, I. Vidal, V. Sanchez-Aguero, F. Valera, and D. R. Lopez, "Software-driven overlay networks for inter-site communications in NFV cross-domains," *IEEE Communications Magazine*, 2022, Submitted in May 2022.

[175] Cloud Native Computing Foundation, *CNCF Survey 2020*, [Online]. Available: https://github.com/cncf/surveys (accessed on May. 17, 2022).

[176] Cloud Native Computing Foundation, *CNI: the Container Network Interface*, [Online]. Available: https://www.cni.dev (accessed on May. 17, 2022).

[177] Flannel, *a simple and easy way to configure a layer 3 network fabric designed for Kubernetes*, [Online]. Available: https://github.com/flannel-io/flannel (accessed on May. 17, 2022).

[178] Tigera, *Calico Open Source*, [Online]. Available: https://www.tigera.io/project-calico (accessed on May. 17, 2022).

[179] Multus, *a container network interface (CNI) plugin for Kubernetes that enables attaching multiple network interfaces to pods*, [Online]. Available: https://github.com/k8snetworkplumbingwg/multus-cni (accessed on May. 17, 2022).

[180] B. Nogales, I. Vidal, V. Sanchez-Aguero, L. F. Gonzalez, F. Valera, and A. Azcorra, *An NFV system to support service provisioning on UAV platforms: a walkthrough on implementation experience and standardization challenges*, Presentation at ETSI NFV Evolution Event, [Online] Available: https://www.telecomtv.com/content/etsi-nfv-evolution-event-agenda-day1, (accessed on May. 17, 2022), 2021.

[181] IEEE, "IEEE 802.1Q - Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks," IEEE Standards Association, IEEE SA, 2018.

Appendix A: Current NFV MANO Platform Specs

This appendix includes the current status of the NFV MANO platform, whose design and implementation principles have been covered in Chapter 3.

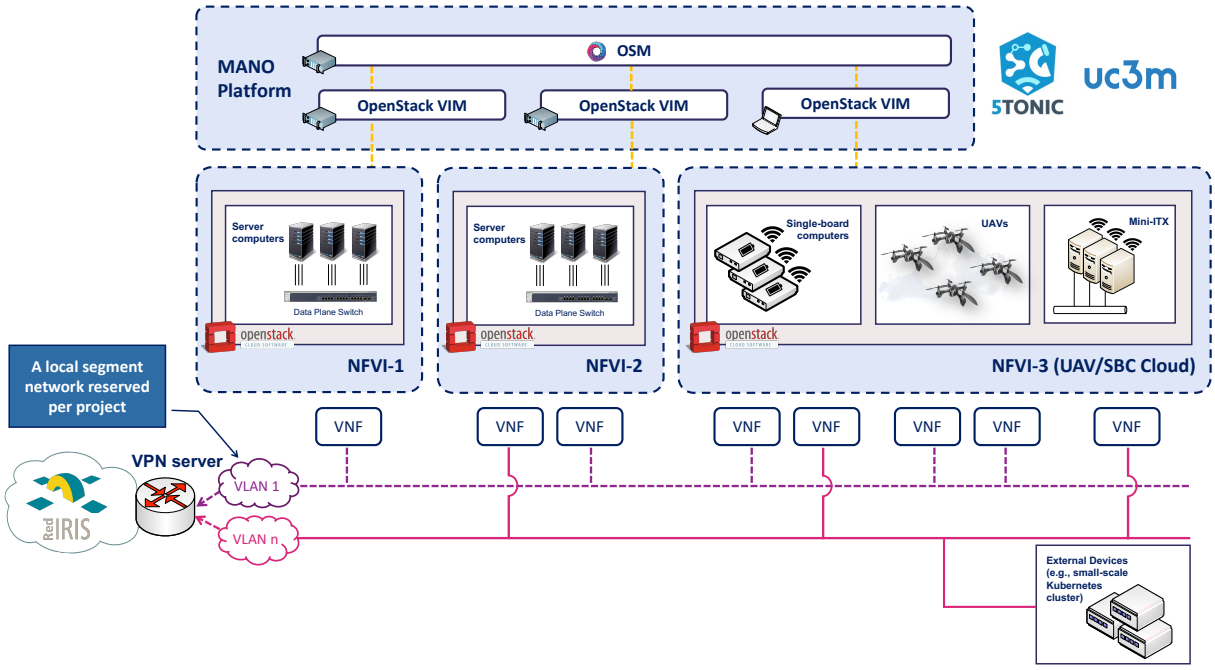


Figure A.1. *5TONIC NFV MANO Platform at present.*

As depicted in Figure A.1, the MANO platform located at 5TONIC is deployed through a set of virtual machines based on OSM and OpenStack. Thus, this setup allows to have three independent NFV Infrastructures (NFVIs) to perform different types of multi-site experiments. In addition, the virtualization of the components comprising the MANO platform allows to easily evolve and/or extend it (*e.g.*, incorporating additional entities to perform experimentation activities with different NFV domains).

In particular, two NFVIs are based on server computers with different computing capabilities. A third NFVI is composed by a set of mini-ITX computers and resource-constrained Single-Board Computers (SBCs). SBCs have a 1 Gbps Ethernet interface and a Wi-Fi adapter, and may be intercon-

nected using different mechanisms according to specific experiment requirements (*e.g.*, forming a wireless ad-hoc network). SBCs behave as mobile NFVI nodes, which can be on-boarded on Unmanned Aerial Vehicles (UAVs) and support experimentation activities beyond the network access segments of telecommunications operators with use cases of aerial networks (see Chapter 4).

In its current state, the platform is configured to support, in an even more flexible manner, the work of Universidad Carlos III de Madrid (UC3M) in different research projects (*e.g.*, 5GZORRO, FISHY, or TRUE5G). To this purpose, each NFVI has been configured to enable the Virtualized Network Functions (VNFs) communications on an isolated network segment depending on the project in which they are deployed. As can be observed in Figure A.1, this division has been realized using OpenStack provider-type networks based on VLANs [181]. This type of networks allows to connect the VNFs to external network appliances (*e.g.*, the VPN server), and to particular physical devices (*e.g.*, a computer device with specific properties provided at hardware layer, or an external small-scale cluster of Kubernetes) that can inter-operate with the VNFs to extend their offered functionalities.

The available computing resources available at each of these three NFVIs are detailed next:

Site 1	**Site 2**	**Site 3 (SUAV/SBC Cloud)**
1 Physical Node (Executing OpenStack Wallaby Controller as a VIM): • Dell server model PowerEdge R630 • 2x Intel Xeon CPU E5-2620 v4 @ 2.1GHz, 8 cores/16 threads • 4x 32GB RDIMM RAM (2400MT/s) • 2x 1TB NLSAS and 2x 2TB • 1x Intel i350 4xGbE with DPDK • 2x 10GbE optical transceivers SFP+ with SR-IOV capabilities	**1 Physical Node** (Executing OpenStack Wallaby Controller as a VIM:) • Dell server model PowerEdge R430 • 1x Intel Xeon CPU E5-2609 v4 @ 1.7 GHz, 8 cores/8 threads • 2x 32GB RDIMM RAM (2400MT/s) • 2x 1TB NLSAS • 1x Intel i350 4xGbE with DPDK	**1 Physical Node** (Executing OpenStack Queens Controller as a VIM): • Commercial Intel Core Mini-ITX Computer • 2x Intel Core i7-3610QE (Ivy Bridge) @ 2.3 GHz, 8 cores/8 threads • 1x 8GB DDR3 RAM (1600 SO-DIMM) • 1x 128GB SSD • 1x Intel 4xGbE

Continued on next page

Site 1	Site 2	Site 3 (SUAV/SBC Cloud)
3 Physical NFVI Nodes (OpenStack Wallaby compute-node): • Dell server model PowerEdge R440 • 2x Intel Xeon Silver 4114 CPU @ 2.20GHz, 10 cores/20 threads • 2x 32 GB RDIMM RAM (2666 MT/s) • 4x 4TB NLSAS • 1x Intel i350 4xGbE ports with DPDK • 1x Broadcom 5720 2xGbE ports • 2x 10GbE optical transceivers SFP+ with SR-IOV capabilities	**3 Physical NFVI Nodes** (OpenStack Wallaby compute-node): • Dell server model PowerEdge R430 • 1x Intel Xeon CPU E5-2609 v4 @ 1.7 GHz, 8 cores/8 threads • 2x 32GB RDIMM RAM (2400MT/s) • 2x 1TB NLSAS • 1x Intel i350 4xGbE with DPDK	**3 Physical NFVI Nodes** (Executing OpenStack Queens compute-node): • Commercial Intel Core Mini-ITX Computer • 1x Intel Core i7-3610QE (Ivy Bridge) @ 2.3 GHz, 4 cores/4 threads • 1x 8GB DDR3 RAM (1600 SO-DIMM) • 1x 128GB SSD • 1x Intel 4xGbE ** These devices support the execution of VNFs in ground equipment within UAV use cases
		3 Physical NFVI Nodes (OpenStack Queens compute-node) : • Raspberry Pi Model 3B+ • 1x Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz • 1x 1GB LPDDR2 SDRAM • 4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN • 1x Gigabit Ethernet port

Continued on next page

Site 1	Site 2	Site 3 (SUAV/SBC Cloud)
		** These SBCs can be onboarded into the SUAVs **3 Physical NFVI Nodes** (OpenStack Queens compute-node) : • Raspberry Pi Model 4 • 1x Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz • 1x 4GB LPDDR-3200 SDRAM • 4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN • 1x Gigabit Ethernet port ** These SBCs can be onboarded into the SUAVs
		SUAVs/Drones: • Parrot Model Bebop 2 • Battery capacity of 2700 mAh • 4 GHz and 5GHz IEEE 802.11a/g/n/ac wireless LAN • 4GHz and 5GHz IEEE 802.11a/g/n/ac wireless LAN • Camera with 14 MP photo resolution, and 1080p30 video resolution • GPS

Continued on next page

Site 1	Site 2	Site 3 (SUAV/SBC Cloud)
		** These vehicles are capable of onboarding a physical NFVI node, such as Raspberry Pi, as payload

Table A.1. *Technical specifications of the current MANO Platform*