

|  |  |  |
| --- | --- | --- |
| Received: 29 April 2018 | Revised: 30 August 2018 | Accepted: 21 December 2018 |

DOI: 10.1002/spe.2681

**SPECIAL ISSUE PAPER**

**SELFNET 5G mobile edge computing infrastructure: Design and prototyping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enrique Chirivella-Perez1 |  |  |  |  | Ricardo Marco-Alaez1 | | | | | |  | Alba Hita1 | | |  | Ana Serrano1 | | | | |  |
|  |  |  |  |
|  |  |  |  |
| Jose M. Alcaraz Calero1 | | |  |  | Qi Wang1 | |  | Pedro M. Neves2 | | | | |  | Giacomo Bernini3 | | | | |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Konstantinos Koutsopoulos4 | | | | |  | Manuel Gil Pérez5 | | |  |  |  | Gregorio Martínez Pérez5 | | | | |  |  | | |  |
|  |  |  |  |  |
|  |  |  |  |  |



Maria João Barros6 Anastasius Gavras6

* School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley, UK
* Altice Labs, SA, Aveiro, Portugal
* Nextworks, Pisa, Italy
* Creative Systems Engineering Monoprosopi Epe, Athens, Greece
* Faculty of Computer Science, University of Murcia, Murcia, Spain
* Eurescom GmbH, Heidelberg, Germany

**Correspondence**

Jose M. Alcaraz Calero, School of

Computing, Engineering and Physical

Sciences, University of the West of

Scotland, Paisley PA1 2BE, UK.

Email: Jose.Alcaraz-Calero@uws.ac.uk

**Funding information**

European Commission Horizon 2020

Program, Grant/Award Number: H2020-ICT-2014-2/671672 ; UWS VP Fund for 5G Video Lab

**Summary**

This paper presents the design and prototype implementation of the SELF-NET fifth-generation (5G) mobile edge infrastructure. In line with the current and emerging 5G architectural principles, visions, and standards, the pro-posed infrastructure is established primarily based on a mobile edge computing paradigm. It leverages cloud computing, software-defined networking, and net-work function virtualization as core enabling technologies. Several technical solutions and options have been analyzed. As a result, a novel portable 5G infras-tructure testbed has been prototyped to enable the preliminary testing of the

integrated key technologies and to provide a realistic execution platform for fur-ther investigating and evaluating software-defined networking– and network function virtualization–based application scenarios in 5G networks.

**KEYWORDS**

deployment, infrastructure, mobile edge computing, network function virtualization, software-defined networks, 5G

* **INTRODUCTION**

While the fourth-generation (4G) or long-term evolution (LTE) networks represent a significant step forward in terms of connecting people and providing advanced services to nomadic customers, the emerging fifth-generation (5G) networks target empowering a fully connected global mobile society to create an unprecedented socioeconomic impact beyond the year 2020. Such an ambitious vision is driving further innovation and underpinning the design and implementation of novel 5G network architectures. As part of the global 5G initiatives, the SELFNET project1 has been launched in Europe under the EU Horizon 2020 5G-PPP (5G infrastructure public-private partnership) program. In strategic terms, 5G net-works are expected to deliver substantially improved performance defined by a set of key performance indicators (KPIs).2 One of the KPIs is to reduce the time required to provision network management services within a 5G network from

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | Softw: Pract Exper. 2020;50:741–756. | [wileyonlinelibrary.com/journal/spe](http://wileyonlinelibrary.com/journal/spe) | © 2019 John Wiley & Sons, Ltd. | 741 | |
|  | *Softw: Pract Exper*. 2019;1–16. | wileyonlinelibrary.com/journal/spe | © 2019 John Wiley & Sons, Ltd. | **1** | |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 742 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **2** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



90 hours to 90 minutes, in order to enhance the competitivity of the telecommunication operators in terms of both capital and operational costs and make their infrastructures agiler against constant business requirements. This KPI has been the main motivation for this research work. It is essential that the architectural design of the SELFNET framework is clearly aligned with 5G architectural visions and compliant with 5G standards under development. In particular, the develop-ment of SELFNET has been in line with the vision of the 5G-PPP,3 the 5G principles from the Next Generation Mobile Networks Alliance,4 and related 5G standards, especially the European Telecommunications Standards Institute (ETSI) mobile edge computing (MEC)5,6 and network function virtualization (NFV).7

The main contributions of this paper are to present the architectural design of SELFNET's 5G MEC infrastructure together with a detailed explanation of how this infrastructure has been technically validated. This paper integrates the following scientific contributions.

* Novel 5G network management architecture where a significant number of technologies, protocols, and standards, including software-defined networking (SDN), NFV, cloud computing, MEC, and so on, are combined and validated in a fully functional infrastructure.
* Highly coordinated automation among all the management planes available in the different layers of the architecture to provide zero-touch orchestration from bare metal.
* Significant extension to the ETSI NFV management and orchestration (MANO) standard to cover the management of the physical machines of the MEC architecture.
* Achievement of the ambitious 5G-PPP KPI on service deployment time under 90 minutes in the proposed architecture, validated through different empirical stress tests.

The remainder of this paper is structured as follows. Section 2 presents an overview of the architectural design of the SELFNET infrastructure, Section 3 focuses on the management plane of the infrastructure, and Section 4 describes the functional validation of the prototype testbed implementation. Section 5 shows the testbed and the empirical results of the 5G service deployment. Finally, Section 6 concludes this paper by explaining the potential development path being considered within the SELFNET consortium.

* **SELFNET INFRASTRUCTURE DESIGN**

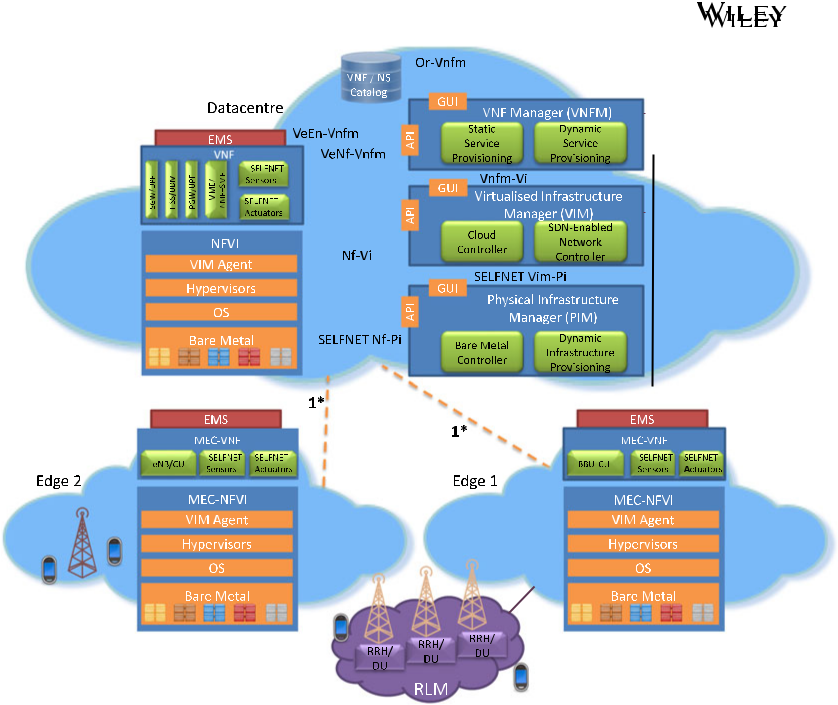
**Mobile edge computing.** The architecture proposed is mainly based on, while significantly extending and improving,the mobile edge architecture envisioned by ETSI. Figure 1 depicts a number of network edges (two of which are shown), geographically separated from the data center. These are used to allocate the operational and management services that need to be deployed close to the user in order to meet the performance requirements. Figure 1 also shows a cloud radio access network (C-RAN) deployment8 in a network edge (EDGE 1). Therefore, Edge 1 controls a pool of geographically dispersed antennas identified in the Figure as Radio Last Mile locations. The proposed does not aim to provide any new 5G air interfaces. This is an area currently being explored by a number of other 5G-PPP projects in Europe9 and by other researchers across the globe.10

**C-RAN.** Some assumptions about the future 5G air interface have been made when designing the proposed mobile edgearchitecture. In particular, it is assumed that the new 5G air interface would be compatible with a C-RAN deployment, which offers two locations where functionalities related to the data and control plane of the new air interface could be placed. This C-RAN approach follows the RAN cloudification trend in 5G and MEC paradigms.

**Optical plane.** Communication links between the edges and the data center (1\* in Figure 1) are expected to be of highdensity with very high data rates and very low latency. In a production environment, this connectivity would be complex and may encompass several technologies, such as wavelength-division multiplexing with reconfigurable optical add-drop multiplexers at the ends of the communication link in order to meet critical 5G KPIs. The architecture does not focus on the data plane; consequently, this aspect has been simplified when prototyping the connection links between different edges. By using commercial off-the-shelf computing equipment for allocated software services in both the data center and network edges, the proposed architecture is expected to significantly reduce capital investment costs. This can be achieved while providing services at the most appropriate location, which may be at the edge closer to the user where it is required to improve the efficiency of a service.

**Internet protocol (IP) networks.** The use of IP networks is assumed. This architectural decision has previouslydriven the evolution from second-generation/third-generation11 to 4G12 mobile networks and will continue being adopted

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  | 743 | |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  | **3** | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**FIGURE 1** Architectural overview of the fifth-generation SELFNET infrastructure [Colour figure can be viewed at wileyonlinelibrary.com]

in 5G networks. Such an assumption is necessary in order to both facilitate the execution of services on top and maximize its impact in the market.

**Operating system (OS).** In order to accommodate the widest range of services, the proposed infrastructure has beendesigned as a heterogeneous OS environment supporting the wide variety of OSs available in the market. Therefore, system architects and service designers will be able to select the most appropriate OS for each service. For example, the allocation of a baseband unit (BBU) service at the network edge may require a real-time–enabled OS, whereas the allocation of services for management purposes may tolerate traditional non–real-time OSs.

**Virtual infrastructure.** The use of virtualization is paradoxical, with significant pros and cons. Therefore, it is impor-tant to consider whether it should be an indispensable or optional component of 5G architectures. The main advantages of virtualization can be defined in terms of enhanced management, reliability, isolation, and control of computational resources. Conversely, the main disadvantage is the performance penalty incurred by introducing this layer. Therefore, a trade-off between functionality and performance may need to be considered when designing 5G architectures. An investi-gation of the state-of-the-art virtualization technologies has been carried out in order to analyze the level of performance penalties that they may incur.

Amit et al13 have recently provided a significant improvement in the data plane of virtualized workloads with intensive input/output required in 5G infrastructures. A throughput penalty in the range from 0% to 3% for intensive input/output applications has been reported together with a latency overhead of around 2% in comparison to bare-metal performance. This performance penalty when using virtualization is largely negligible for modern servers and is outweighed by the benefits of virtualization, which justifies our employment of virtualization technologies in the proposed infrastructure.

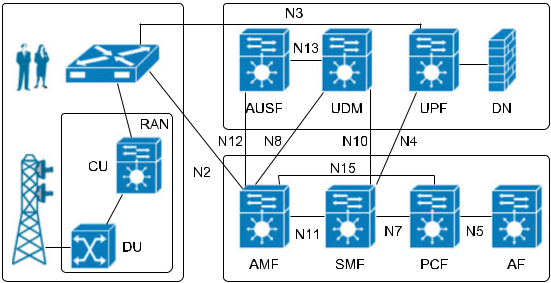
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 744 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **4** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



It is worth noting that, in order to support a wide range of use cases, the proposed infrastructure is able to deal with both virtualized and bare-metal service deployments (without virtualization). Furthermore, support for heterogeneous virtualization technologies may be required within the 5G architectures managed on top of the infrastructure. It is impor-tant to note that, regardless of the specific virtualization technology being used, the usage of virtualization implies the use of virtual switches implemented in software. These are to interconnect different virtual machines (VMs) allocated within the same physical machine. Different hypervisor technologies have been analyzed using well-known hypervisors, such as kernel-based virtual machine (KVM),14,15 Quick Emulator (QEMU),16,17 VMWare,18,19 Linux Containers (LXC),20 VirtualBox,21 and XEN.22 As a preliminary result of this analysis, KVM has been recommended as a promising candi-date to achieve hardware-based virtualization that meets the required performance levels. In addition, LXC has been recommended when kernel-based light virtualization providing solutions suitable for supporting software with real-time requirements is needed. Management of the virtualization layer provides multitenancy support over the virtual infras-tructure by smartly configuring virtual switches and VMs. The use of a virtual infrastructure enables the deployment of virtual topologies. By adopting an SDN approach, the management of the different network segments and the control of the connection of VMs to a given network segment can also be enabled, thereby giving the functionality needed to create any potential topology required in 5G architectures and enabling the self-management of such virtual topologies using the proposed infrastructure.23

**4G/5G infrastructure.** Figure 1 depicts an overview of a basic LTE (including LTE Advanced) architecture, its currentmain architectural components, and how they are deployed across different locations within the network. Initially, LTE has been employed to build the SELFNET framework; although, going forward, the project will track the continuous evolution from LTE to 5G networks (see Figure 2) and continue to ensure that the framework remains aligned with ongoing 5G developments. Conceptually, LTE is divided into a control plane and data planes. LTE24 has the following components.

* **Authentication Server Function.** This function is part of the Third Generation Partnership Project (3GPP) 5Garchitecture, which is used to facilitate 5G security processes.
* **Unified Data Management.** This component is related to the 3GPP 5G architecture, supporting the AuthenticationCredential Repository and Processing Function to store long-term security credentials used in the authentication for Authentication and Key Agreement. In addition, it stores subscription information.
* **Core Access and Mobility Management Function.** This function is part of the 3GPP 5G architecture. Its primarytasks include Registration Management, Connection Management, Reachability Management, Mobility Management, and various functions related to security and access management and authorization.
* **Session Management Function.** This function is related to the 3GPP 5G architecture and is one of the main functionsin the Next-Generation Core. As such, it includes various functionalities relating to subscriber sessions, such as session establishment, modification, and release.
* **Policy Control Function.** The Policy Control Function is related to the 3GPP 5G architecture. This function supportsthe unified policy framework that governs network behavior. In so doing, it provides policy rules to control plane



**FIGURE 2** Fifth-generation reference architecture [Colour figure can be viewed at wileyonlinelibrary.com]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | 745 | |  |
|  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | **5** | |  |
|  |  |  |  |  |  |  |



function(s) to enforce them. In order to facilitate this, subscription information is gathered from the Unified Data Management function.

* **Application Function.** The Application Function is a logical element of the 3GPP Policy and Charging Control frame-work, which provides session-related information to the Policy and Charging Rules Function in support of Policy and Charging Control rule generation.
* **User Plane Function (UPF).** The UPF is related to the 3GPP 5G architecture. It is similar to the roles played bythe Serving/Packet Gateway in a 4G LTE system. The UPF supports features and capabilities to facilitate user plane operation. Examples include packet routing and forwarding, interconnection to the DN, policy enforcement, and data buffering.
* **Data Network (DN).** The DN is related to the 3GPP 5G architecture. It identifies Service Provider services, Internetaccess, or third-party services.

Any user equipment gains access to the network via a connection to the antenna, which is controlled by the E-UTRAN/Evolved Node B or, in a C-RAN environment, the BBU+Radio Remote Head whose functionalities cover con-trol of the air interface and provision of connectivity to both data and control planes of the mobile network. A more comprehensive description of the different architectural components available in LTE and their architectural relationship with the SELFNET project can be found in the work of ElNashar et al.25 It is also worth noting that LTE infrastructures currently provide a number of mechanisms that enable LTE architectural components to be shared by different mobile operators. Several open-source and closed implementations were analyzed in order to inform our choice of the software stack to be used for prototyping purposes. These included the LTE-EPC (Evolved Packet Core) Network simulator LENA26 running in the Network Simulator (NS-3),27 OpenAirInterface (OAI),28 and a number of proprietary solutions. Multite-nancy capability support in the antenna, known as the Multi-Operator Core Network (MOCN),29 has been standardized by the 3GPP.30 It is expected that 5G architectures will provide similar, or enhanced, mechanisms for sharing architectural components in a multitenancy environment. The main intention of multitenancy provisioning is to significantly reduce both capital and operational expenditures (CAPEX and OPEX, respectively).

**SDN controller.** The proposed infrastructure decouples the data and control planes of the network and, in doing so,provides a logically centralized control element capable of governing the complete set of devices available in the network. This controller (henceforth referred to as the SDN controller) has a holistic view of the network, enabling it to enforce the set of actions that need to be performed in the network elements in order to correctly handle a new data flow passing through the network. This centralized management requires the use of SDN-enabled network elements capable of being configured by an SDN controller. Importantly, the centralization of control of the network does not imply a central point of failure or a bottleneck in terms of performance. In fact, modern SDN controllers use clustering approaches, high avail-ability, and other methods to ensure their scalability and performance and to enforce governance. The SDN controller underpins the network management and governance described in Section 3.

* **NETWORK MANAGEMENT ON 5G SELFNET INFRASTRUCTURE**

In SELFNET, great importance is attached to the reduction of provisioning time for new services. Across a range of use cases, automation is used as a key enabler to reduce provisioning time. This is accomplished in a range of scenarios where

1. new hardware is being inserted, replaced, or removed from the infrastructure; (2) new services are being deployed, undeployed, or redeployed; (3) support for the management of virtual infrastructures is inserted; (4) configured and con-trolled; (5) virtual infrastructures are created, destroyed, and migrated; and (6) virtual services are deployed, undeployed, or redeployed in virtual infrastructures. Figure 1 additionally depicts our vision for the management plane of the physi-cal layer of MEC infrastructures for 5G architectures. The vision is completely aligned with the standardized ETSI NFV MANO architecture,31 while providing significant extensions and improvements to the standard. In the interest of clarity, the naming of SELFNET components has been aligned with that of the ETSI NFV MANO standard.32

**Physical infrastructure**. The proposed architecture considers a number of architectural considerations, for example,management of multiple geographically separated physical locations, ie, edges and data centers. Multitenancy hardware resource sharing among telecommunication operators (telcos), energy monitoring, and management of network hard-ware and automated OS and software installation are provided by the SELFNET Physical Infrastructure Manager (PIM). SELFNET also enables the management of the physical infrastructure, from bare metal, through Cluster Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 746 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **6** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



Configuration and Provisioning Tools, such as Metal-as-a-Service (MaaS),33 Rocks Cluster,34 and HP Cluster Management Tool.35 The PIM also provides the business logic for management and synchronization of the physical infrastructure.36 This architectural element significantly optimizes provisioning time and enhances the reliability and availability of the OS provisioning service for managed nodes.

**Energy management.** One of the targets of the use cases proposed in SELFNET is to reduce end-to-end energy con-sumption. To do so, we need to firstly meter the energy consumed by all elements involved in the end-to-end provision of a service, starting from the wireless access network with its physical antennas and continuing with the computing resources that will be used to execute the software components in the virtual infrastructure. Two different approaches can be taken to monitoring energy usage: out-of-band interfaces or in-band interfaces. Open Energy Monitor37 is an interesting example of an out-of-band tool, an open-source monitoring system based on Arduino38 that measures electricity con-sumption. This tool can be combined with SEGmeter,39 another Arduino-based tool. eNOS40 is a cloud-based open-source energy management system that measures electricity, consumption, etc, and provides a toolkit to build a cloud-based energy dashboard. The architecture has to provide a mechanism by which physical network resources can be remotely powered up or instructed to hibernate to meet the energy usage targets of the network. The Intelligent Platform Manage-ment Interface mechanism provides power management facilities by communicating, through a dedicated management network interface, with the Baseboard Management Controller. In-band power management can be also achieved using the Wake-on-LAN mechanism. This legacy approach offers the ability to send control packets to the Network Interface Card (NIC), which, in turn, signals the power supply unit or motherboard to start/up shutdown. Wake-on-LAN uses MAC addressing rather than IP addresses. It lacks the functionality in remote access to Basic Input/Output System and Uni-fied Extensible Firmware Interface and the flexibility in out-of-band and sideband connectivity of the Intelligent Platform Management Interface.

**Service provisioning.** Once the physical infrastructure is provisioned with an OS, the infrastructure is ready for theprovisioning of the software services. In fact, the same approach can also be adopted when a virtual infrastructure is already provisioned with an OS. The process of customizing an image to provide a specific service can be performed following a static or a dynamic approach. The static approach requires an image, which contains the appropriate software preconfigured, ie, NVF software that it starts automatically after booting. The dynamic approach relaxes these constraints and allows the image to be customized after being started. The static approach to customization of volume images depends on the operations staff to create a volume image prior to deployment, in which the OS and all services to be provided are already installed and properly configured according to the requirements of the organization. This approach creates standardized images that can be reused to deploy similar variants of the service but has many limitations. Firstly, the operations staff needs to perform the installation and configuration of all required services manually. Secondly, images must be maintained on a regular basis, applying patches for security and other reasons. Thirdly, each time the image is changed, the new version has to be uploaded. This might be very time consuming since a typical volume can easily be many gigabytes in size and must be transferred over the network. Finally, it is difficult to offer flexibility in configuration, since every configuration option leads to a possible new image that needs to be created, uploaded, and maintained by the operations staff. The dynamic approach has also advantages and disadvantages. Firstly, everything could be performed automatically, reducing the time to provision the services. Secondly, only the required software packages, rather than the whole volume image, need to be sent over the network, making the deployment process much agiler. Finally, the maintenance of the base volume image is easier since there are fewer base images. The main disadvantage is that the required time to boot an image with respect to a ready-to-run image previously configured using a static approach is significantly higher. Both approaches have been carefully analyzed in the proposed architecture, resulting in a decision to support both approaches in order to offer flexibility in deciding which one is better for each use scenario and to allow a combination of what are essentially complementary approaches. The static provisioning of services can be provided by tools like Ghost,41 Clonezilla,42 PartImage,43 or FOG44 at a physical layer, or by the virtualized file system provided by almost all the hypervisors available in the market, such as Virtual Disk Image,45 Virtual Machine Disk,45 and QEMU Copy on Write,46 to name a few. On the other hand, dynamic provisioning is usually performed by configuration management tools. There are a number of tools of this type available in the market, such as Puppet,47 CHEF,48,49 SmartFrog,50,51 Juju,52 and Ansible.53,54 These tools enable the automatic deployment and configuration of software services within the virtual and physical infrastructures. Figure 1 illustrates these two deployment capabilities together with the control of the life cycle of the deployed services by means of the Virtual Network Function (VNF) Manager (VNFM) and its correspondent interfaces to the VNFs and Element Management System (EMS). The implementation of this automatic deployment of services is another important step in the reduction of the time required to deploy network services within the new 5G architecture.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | 747 | |  |
|  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | **7** | |  |
|  |  |  |  |  |  |  |



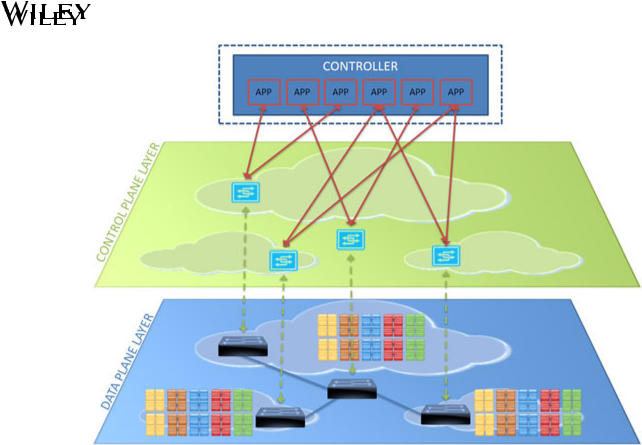
**Virtual infrastructure**. The virtual infrastructure proposed offers virtualized resources (network, computing, andstorage), cloud-based multitenant support, monitoring55 and management of virtual resources, and VNF deployment sup-port. The management of the virtual infrastructure is achieved through Infrastructure-as-a-Service.56,57 Figure 1 shows the Virtual Information Manager (VIM) made up of an Infrastructure-as-a-Service stack, such as OpenStack,58 CloudStack,59 and OpenNebula,60 and an SDN-enabled network controller, such as OpenDaylight (ODL),61 Open Network Operating System (ONOS),62 Ryu,63 and FloodLight.64 The VIM provides Authentication Services, Image Management Services, Block Storage Services, Certificate Management Services, Scheduling Services, and a Networking Management Service. The Networking Management Service ensures that the network is not a bottleneck or a limiting factor in a virtual infrastructure deployment and gives users self-service capabilities over network configurations. Multitenancy support for virtual infrastructures is one of the main capabilities achieved by the management plane of the virtual infrastruc-tures. It enables tenants to share the same physical infrastructure in a completely isolated way where all virtual resources are isolated and protected between tenants. All components described above are able to provide multitenancy capabili-ties ranging from the GUI and APIs offering information only to the relevant tenant, to complete isolation of the virtual networks (shared in the physical plane) by different networking technologies, eg, Virtual Local Area Network,65 Virtual Extensible Local Area Network,66,67 Generic Routing Encapsulation,68,69 as well as other encapsulation technologies. This multitenancy should now be extended to the edges of the networks.

**SDN controller**. The SDN infrastructure proposed employs an SDN controller–centric paradigm to link thebusiness-specific SDN-Apps and the underlying virtualized network elements. Key technologies investigated include, among others, Virtual Tenant Network,70 Service Function Chaining (SFC),71 Network Slicing,4 and multitenancy support for SDN-Apps. An approach being primarily considered nowadays is the use of an SDN controller as a logically centralized controller, where the control of the network functions and the governance of the different network elements managed in the infrastructure can be implemented. A logically centralized approach for management purposes enables the com-position of holistic views of the network, and thus, it enables the creation of advanced network intelligent protocols to control the traffic in the network, having wider information about the current status of the network provided by all net-work elements controlled. The main idea is to configure all network elements to be controlled by such SDN controllers. Then, when the traffic is passing through such network elements, the SDN controller will enforce the rules being config-ured in these network elements. The initial configuration and periodical updates of configuration will be conducted by the SDN controller dynamically, according to the protocols running in the SDN controller. Below, the reader can find a list of key functionalities considered in the proposed architecture that may be provided by SDN-Apps for the 5G mobile edge infrastructure.

* **Virtual Tenant Network** isolation manages virtual networks (Layer 2 or L2 domains) and virtual ports connectedto such virtual networks in order to provide true L2 slicing and traffic isolation of the traffic associated to each of the tenants. The practical implementation of this functionality could be based on tunneling and tagging protocols such as Multiprotocol Label Switching,72,73 Virtual Local Area Network, Virtual Extensible Local Area Network, Generic Routing Encapsulation, etc. The aim is to achieve the manipulation of logical maps of the network and create multiple coexisting virtual networks for each tenant regardless of the underlying transport technology and network protocols.
* **Network slicing**, or, more precisely, infrastructure slicing in the context of this document, enables end-to-end virtualinfrastructure sharing for multitenants. Fifth-generation network infrastructures will become increasingly multi-tenant, hosting heterogeneous types of tenants with different requirements, with the need of adopting a service model where each tenant is provided with its own virtual infrastructure. In this context, network virtualization becomes crucial and, when combined with IT virtualization, allows service providers to create multiple coexisting isolated end-to-end virtual infrastructures for their tenants.
* **SFC**71,74,75defines a specifically ordered list of network services/functions (eg, load balancers, firewalls, etc) based onuse scenario requirements and manages traffic redirection functionalities to ensure that the traffic in the infrastructures is flowing through certain services.
* **Security Group Control**76,77manages firewall-filtering functionalities in order to provide control of traffic being pro-duced or consumed in the infrastructure. It includes filtering of traffic based on MAC and IP addresses, port number, packet type, etc.

Figure 3 shows an overview of the intersection between the data and control planes of the network. The bottom part presents the data plane. The introduction of a central switch logically connecting all edges has been considered (especially

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 748 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **8** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



**FIGURE 3** Overview of the architecture for the management of the control plane in mobile edge computing infrastructures forfifth-generation architectures [Colour figure can be viewed at wileyonlinelibrary.com]

for the prototype testbed implementation). It will enable the SDN controller to have an enforcement point where other overlay services and capabilities can be added into the network in order to control traffic flows. These functionalities have been identified in SELFNET Deliverable 2.178 as a wide-area network controller. It has been decided to consider this option so as not to limit the architecture vision. The inclusion of this type of SDN-App for controlling the wide-area network traffic would be decided later according to the requirements of the project toward the implementation of the use cases.

Figure 3 depicts an overview of the architecture deployed for the management of the control plane in MEC infrastruc-tures in 5G architectures. In the control plane shown in the upper part of Figure 3, the controller is logically centralized in the data center. However, it is important to emphasize that a logically centralized SDN controller does not mean that it would be, architecturally speaking, a single point of failure or a potential bottleneck in the network. The SDN con-troller can be implemented in a highly distributed way and offer a logically centralized abstraction for the control of the network. The SDN controller has functional elements for such a distribution allocated in each of the edges and the data center. Most SDN solutions are based on powerful network abstraction approaches that allow controlling and provisioning heterogeneous technologies in a unified way.

The extensions to the ETSI NFV standard proposed in this architecture are defined as follows.

* The extension of the management plane to include the new PIM architectural component in charge of the management of the life cycle of physical machines in the MEC infrastructure.
* A new interface between PIM and VIM to allow the management plane to provide homogeneous understanding of the different geographical zones available in the MEC infrastructure so that there is an alignment between the geographical zones of both physical and virtual computers.
* A new orchestrator that encloses wider responsibilities than those defined by the NFV Orchestrator existing in the ETSI MANO standard. This new orchestrator takes the responsibility of deploying services from bare metal with a zero-touch interaction with the infrastructure. This zero-touch innovation allows network administrators to saved significant time in the service deployment.

It is noted that our proposal realizes an integrated multilayer network MANO architecture of 5G networks. Table 1 sum-marizes the technologies adopted in the validation of the proposal in this paper for the various functionalities required, across the different layers including physical, virtual, and service layers, to provide a complete solution.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. | |  |  |  |  |  |  | 749 | |  |
|  |  |  |  |  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. | |  |  |  |  |  |  | **9** | |  |
|  |  | | | |  |  |  |  |  |  |  |
|  | **TABLE 1** Functionalities and technologies across architectural layers | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | **Physical Layer** | **Virtual Layer** | **Service Layer** | |  |  |  |  |  |
|  |  | **Physical Infrastructure Management** | MaaS | MaaS | – | | | | | |  |
|  |  | **Service Infrastructure Management** | Juju | Juju | Juju | |  |  |  |  |  |
|  |  | **Hypervisor** | QEMU | KVM | LXC/LXD | | | | | |  |
|  |  | **Networking** | OpenvSwitch | OpenvSwitch | OpenVSwitch | |  |  |  |  |  |
|  |  | **SDN Controller** | OpenDaylight | OpenDaylight | OpenDaylight | | | | | |  |
|  |  |  |  | SFC | SFC | | | | | |  |
|  |  |  |  | VTN | VTN | | | | | |  |
|  |  |  |  |  |  |  | |  |  |  |  |
|  |  | **Energy Management** | Open Energy Monitor | – | – | |  |  |  |  |  |
|  |  | **Monitoring** | NFVMon | NFVMon | – | | | | | |  |
|  |  | **LTE** | SDR Ettus x310 | OpenAirInterface | – | |  |  |  |  |  |
|  |  | **Virtual Infrastructure Manager** | – | Openstack | Openstack | |  | | | |  |



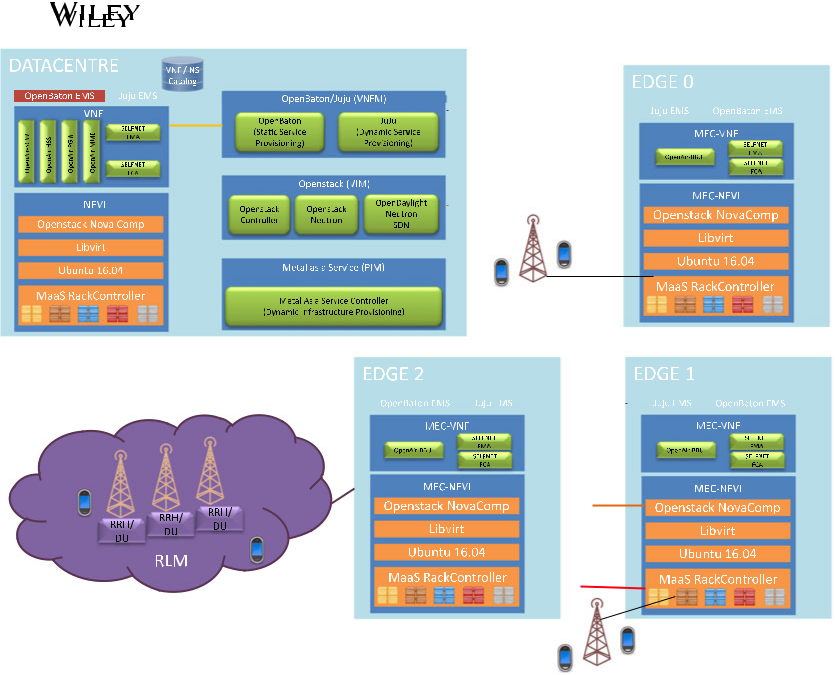
Abbreviations: KVM, kernel-based virtual machine; LTE, long-term evolution; LXC/LXD, Linux Containers; MaaS, Metal-as-a-Service; NFV, network function virtualization; QEMU, Quick Emulator; SDN, software-defined networking; SFC, Service Function Chaining; VTN, Virtual Tenant Network.

* **FUNCTIONAL VALIDATION**

There are a significant number of innovations indicated in this paper, and preliminary empirical validations of the essen-tial innovative designs have been conducted. The following list summarizes the prototypes that have been validated through implementation and functional demonstration to validate the technical approaches adopted in different com-ponents of the architecture presented in previous sections. These prototypes are shown in the infrastructure depicted in Figure 4.

* **Physical Infrastructure Manager.** This research has developed a customized version of MaaS33to enable the deploy-ment of a PIM suitable to control computational resources allocated in different geographical locations and to facilitate the deployment of the physical infrastructure in the data center and the edges of the network. The extension has been focused on providing MEC capabilities over MaaS. An initial support for MEC capabilities was already available in MaaS. However, MaaS has a strong requirement related to the assumption that all the nodes deployed within the same geographical zone have access to the Internet, whereas in 5G architectures, this is not normally the case, and there is only connectivity to the core network segment. This limitation has been overcome in our prototype by extending MaaS. The MaaS Rack Controller in all the edges of the infrastructure and MaaS Controller in the Data Center can be found in Figure 4.
* **Virtual Infrastructure Manager.** The prototype employs all the services related to the cloud controller and the SDNcontroller to set up multilocation multitenancy scenarios. OpenStack has been used as the VIM and OpenDaylight as the SDN controller. It has achieved a complete integration between PIM and VIM in order to enable truly multilocation capabilities in OpenStack. This integration has required the creation of a new component that integrates the different geographical zones managed in MaaS with the different availability zones controlled by OpenStack. See OpenStack Nova Compute in all the edges of the infrastructure and OpenStack Controller and OpenStack Neutron in the Data Center in Figure 4.
* **Networking and SDN control services.** Networking Gateway capabilities and the SDN controller have beenprototyped. Three different prototypes have been implemented based on Neutron, OpenDaylight+Neutron, and ONOS+Neutron. Finally, OpenDaylight has been used and recommended due to its maturity with respect to the Neutron northbound interface provided with SFC capabilities. See OpenDaylight Neutron SDN in Figure 4.
* **Static service deployment.** Two different prototypes have been implemented to deploy a VNFM implementation incharge of performing static provisioning of VNFs into a VIM infrastructure with multilocation support. The prototypes are based on OpenMANO and OpenBaton; both have been successfully integrated into OpenStack. At the moment of the prototyping, none of them provides real support to deal with multiple availability zones efficiently. Finally, OpenBaton has been used and recommended due to the architectural design of the software that makes it easier to include the new capabilities into the VIM driver of the OpenBaton architecture. See OpenBaton in the Data Center and OpenBaton EMS in each of the edges in Figure 4.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 750 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CHIRIVELLA-PEREZ ET AL. | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CHIRIVELLA-PEREZ ET AL. | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**FIGURE 4** Prototyped fifth-generation mobile edge computing (MEC) infrastructures [Colour figure can be viewed atwileyonlinelibrary.com]

* **Dynamic service deployment.** The prototype deploys a VNFM implementation to perform dynamic provisioningof the VNFs over both physical and virtual infrastructures.79 Three different prototypes have been implemented based on Juju, CHEF, and Puppet, respectively. Juju offers promising capabilities due to its integration with both OpenStack and MaaS, which allows also the control of the location where the VNF is located. Thus, Juju has been used and recommended due to its natural integration with both MaaS and OpenStack. See Juju in the Data Center and Juju EMS in each of the edges in Figure 4.
* **4G/5G services.** The prototype deploys an LTE infrastructure by means of the VNFM manager integrated with bothVIM and PIM into an MEC architecture. The infrastructure is fully virtualized, and it has been integrated within OpenStack using different virtualization technologies, concretely, KVM and Docker/LXC. The LTE infrastructure has been prototyped using OpenAirInterface VNFs, composed by HSS, MME, and SGW/PGW running in the Data Center. These are using KVM and BBUs operational in each of the edges running in LXCs with hardware devices USBP B210 and Mobile Phones LG directly connected to these LXC containers. See the VNFs for the 4G/5G services in Figure 4.

The prototypes have created a proof-of-concept implementation. Two different versions have been implemented. One version has only one edge location as a first step, enabling the consortium to run services. The other version is an extension in which another edge is also made available. Each of the geographical locations comes with compute nodes where VMs can be allocated.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | 751 | |  |
|  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | **11** | |  |
|  |  |  |  |  |  |  |



There are other works that investigate the automatic process to reduce times from various perspectives. A related sce-nario is that the deployment time to create new operators needs to be reduced since the mobile virtual network operators exist. For instance, Martínez et al80 addressed a Multi-Layer (Packet and Optical) Aggregation Network and proposed SDN/NFV orchestration to compose isolated backhaul tenants used for different mobile virtual network operators. How-ever, no numerical results on time reduction are reported. Moreover, a more relevant aspect is to deploy new network elements/functions in the SDN/NFV context and manage to reduce the deployment time. For example, Katsalis et al81 proposed to explore a network slicing architecture for 5G communications and presented some indicative results on deploying an LTE Evolved Node B as a VNF using Juju over a clean installation. However, no complete and integrated solution that is able to reduce the times in all the layers is found in the literature. Controlling and having an absolute control of the life cycle in all the layers is the solution that our work achieves.

* **TESTBED AND EMPIRICAL RESULTS**

In order to validate the proposed infrastructure against the ambitious 5G KPI related to the fast deployment of 5G services, complete automated orchestration of PIM, VIM, and SIM has been implemented to achieve a fully automated deploy-ment of services into virtualized infrastructures from bare metal. To this end, when a new physical machine is attached to our edge, it is detected and automatically configured to enable the power management of this physical machine to gain control over the hardware resources. Then, MaaS is invoked to perform the installation of the operating system, and later on, Juju is called, in order to install OpenStack compute service on top of such physical machine to allow VIM to acquire control of the hardware resources. Subsequently, Juju is re-invoked to act now on top of OpenStack to perform the deployment of the 5G virtual services, ie, OpenAirInterface VNF component, on the new hardware resources con-trolled by OpenStack. All these steps contribute to the overall service creation (deployment) time of 5G services from bare metal.

The testbed has been created using eight physical machines as managed computers, each one having 8 cores, 24 GB of RAM, and 2x1 Gb/s Ethernet NICs. These machines are managed by a physical machine with an Intel Xeon Processor E5-2630 v4 with 32 GB and 1x10 Gb/s Ethernet NICs acting as a management plane. The purpose of this testbed is to empirically investigate the service deployment time consumed to perform the installation of virtual services for 5G infras-tructures from bare metal. To this end, the physical machines have been virtualized to emulate a larger infrastructure with our current physical resources. Consequently, eight VMs have been created on each of the six physical computers, resulting in 48 virtualized physical machines in total, by utilizing nested virtualization when the VMs are created on top of OpenStack. Since the main purpose is not to optimize the performance of the virtualized service deployed, but to demon-strate the scalability of the proposed architecture, this deployment has allowed us to perform the deployment of a larger number of hardware resources. Thus, the number of VMs is ranged from 1 to 48.

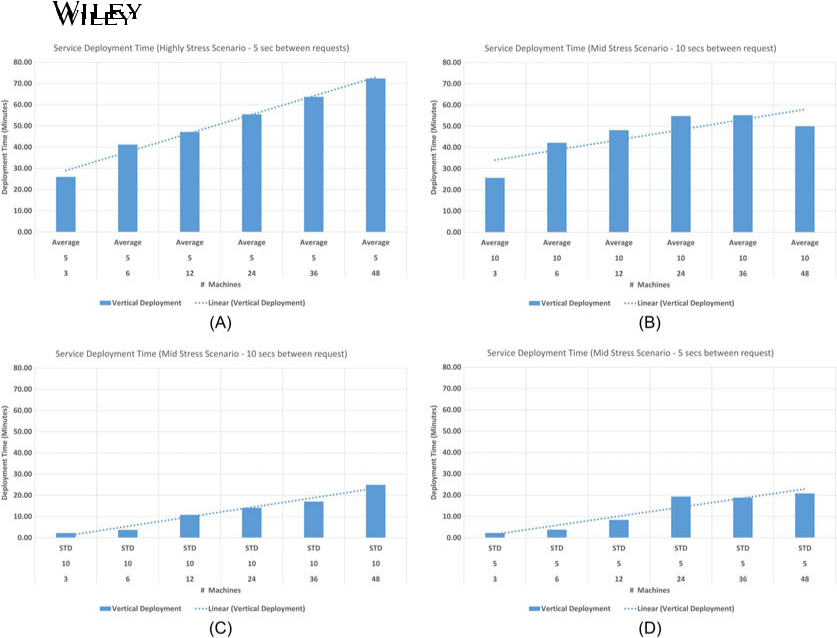
The 48 VMs are bare-metal VMs, and at the end, all of them should be controlled by OpenStack, and a virtualized service for monitoring 5G networks should be deployed in each of them. Between the execution of each of the scenarios to be analyzed later in this section, a complete clean-up of the VMs was performed in order to allow that the execution of the experiment was always from bare metal. For each of the scenarios where a given number of physical machines were provisioned, two different ramping times have been analyzed.

The ramping time is defined as the time elapsed between the moment when a new VM is started into the infrastructure and the moment when the previous VM was started. A ramping time of 0 means the parallel starting of all the VMs at the same time. In addition, a realistic high-stress scenario with a ramping time of 5 seconds and a realistic mid-stress scenario with a 10-second ramping time were applied, which means that one new VM is connected to the infrastructure per 5 or 10 seconds, respectively. Subsequently, the VMs were deployed using a vertical orchestration strategy where all the VMs allocated in the same physical machine are deployed before starting the deployment of further VMs in the next one.

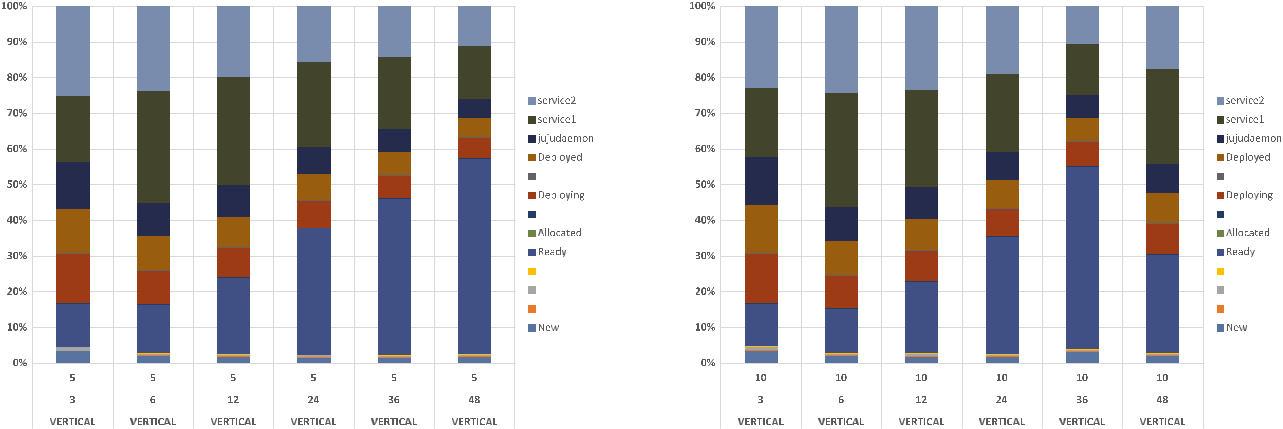
It is worth noting the significant time-consuming aspect related to the execution of the experiment, and the service creation times are presented in terms of minutes. Ten different executions of each of the scenarios have been carried out, and the averaged times are those plotted in Figure 5. Once the experimentation has been designed, implemented, and prototyped, it has required significant execution time (10 executions × 2 experiments × 6 scenarios × 90 minutes approximately) to gather the results presented herein.

Two main experiments have been carried out, where the 5-second (left side of Figure 5) and 10-second (right side of the Figure) ramping times have been tested and analyzed, respectively, to investigate the behavior of the proposed architecture

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 752 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **12** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



**FIGURE 5** Fifth-generation service creation time from bare metal. A, Average time for 5-second ramping time; B, Average time for10-second ramping time; C, Standard deviation for 5-second ramping time; D, Standard deviation for 10-second ramping time [Colour figure can be viewed at wileyonlinelibrary.com]



(A)

(B)

**FIGURE 6** Fifth-generation service creation time breakdown. A, % Percentage time consumed with 5-second ramping time;B, % Percentage time consumed with 10-second ramping time [Colour figure can be viewed at wileyonlinelibrary.com]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | 753 | |  |
|  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | **13** | |  |
|  |  |  |  |  |  |  |



at different stress levels. Figure 5A,B shows the average execution time in each experiment, whereas Figure 5C,D shows the standard deviations associated with each experiment. The *y*-axis values of the plots represent the minutes to perform the deployment. The starting time is measured when the first network packet is detected in the experiment, usually related to the packet exchange involved in the Preboot Execution Environment booting protocol. The ending time is defined as the time when the deployment of the last service of the last VM is completed and the service becomes up and running. The *x*-axis values of the plots show the different scenarios that were executed in the testbed, specifically the number of VMs involved in each of the scenarios. The results shown are the average of all the times gathered per physical machine. Both *x* and *y* axes are represented using an exponential distribution, and the size of the bars in the Figure follows a linear distribution, which is the base to consider the architecture scalable. Furthermore, it shows a slope with an acceptable gradient, which is a significant achievement in the factor of scalability of the proposed architecture. It should be noticed that the standard deviations along all the service deployment times achieved along all the machines involved in each of the experiment also show a very slow gradient, which is a clear sign of the stability and resilience of the proposed infrastructure in the context of scalability in terms of the number of nodes.

Figure 5 also shows that the differences in the service creation times between the different scenarios that involve differ-ent ramping times are insignificant. The results demonstrate good responsiveness against high-stress conditions where the ramping time is reduced to half, which is directly related to a double-stress condition. Moreover, it should be noticed that, in all the scenarios analyzed in terms of size and stress conditions, the service creation times, on average, and taking into account the standard deviation fulfill the ambitious KPI set by the EU 5G-PPP program, ie, creating a new 5G service even from bare metal in less than 90 minutes.

Figure 6 shows the percentage of the average time consumed across the different steps involved in the deployment of a service from bare metal. Figures 6A and 6B provide an analysis of the behavior in the distribution of the average time consumed for scenarios with ramping times of 5 and 10 seconds, respectively. As can be observed, the distribution in times is not affected significantly by the ramping time, and it is clear from the analysis of the graphs that the main contribution to the overall times comes from the “Ready” phase. This time represents the time between the moment where the machine is ready to be used and the time where the machine is selected by the Juju scheduler in order to be used as the target for the deployment of the service.

* **CONCLUSIONS**

The proposed 5G MEC infrastructure has been designed and prototyped in a realistic testbed implementation. Architec-tural decisions have been taken, wherever appropriate, in order to align the proposed infrastructure with the latest and most innovative trends in the control, management, and data planes of softwarized 5G networks. The architecture pre-sented is flexible and extensible, which allows it to cope with the architectural evolutions foreseen from other 5G research activities. Moreover, it is noted that the proposed infrastructure is agnostic to the 5G air interface design, which is an ongo-ing work both within the EU and globally. Comprehensive design considerations for the data, control, and management planes have been presented, centered on the MEC architecture.

This research has employed OpenStack for implementing the VIM. No existing automation tool has been able to provide a complete deployment of OpenStack integrated with OpenDaylight or any other SDN controller, as accomplished in this research. Furthermore, it is worth highlighting that a completely functional LTE infrastructure running in virtual infrastructure has been achieved. Both the real hardware mode using standard mobile phones and an emulation mode using software phones have been enabled to facilitate experimentation over a portable infrastructure. All the software used is based on open-source implementations. It is noted that the proposed infrastructure is not constrained to a specific SDN controller. Two promising SDN controller candidates that offer SFC capabilities, ONOS and ODL, have been installed and analyzed, and a functional demo of SFC capabilities in ODL has been achieved.

It is noted that all the major design aspects proposed in this document have been implemented in the prototype testbed. This research has experimentally achieved noticeable innovation toward a novel 5G infrastructure design and implemen-tation. In particular, significant achievements have been made and empirically tested in the prototype testbed to reduce the service creation time in physical and virtual infrastructures, motivated by meeting the ambitious KPI in substantially reducing service creation time envisioned by the 5G-PPP association and the European Commission. An empirical valida-tion of the achievement of reducing service creation time from 90 hours to 90 minutes has been conducted. The scalability of the architecture and the resilience against the size of the infrastructure have been empirically validated, tested, and

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 754 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **14** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



analyzed by means of intensive testing and in all the executions to demonstrate that the concerned 5G-PPP KPI has been achieved through fully automated service deployment introduced by this research.

**ACKNOWLEDGEMENTS**

This work was, in part, supported by the European Commission Horizon 2020 Program under grant H2020-ICT-2014-2/671672—SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). This work was also supported by the UWS VP Fund for 5G Video Lab.

**ORCID**

*Jose M. Alcaraz Calero* https://orcid.org/0000-0002-2654-7595

*Anastasius Gavras* https://orcid.org/0000-0003-4496-8358

**REFERENCES**

1. 5G-PPP. SELFNET Project. https://5g-ppp.eu/selfnet/
2. 5G-PPP. Key performance indicators (KPIs). https://5g-ppp.eu/kpis/
3. 5G-PPP. 5G vision. https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf
4. *5G White Paper*. Frankfurt, Germany: NGMN; 2015. https://www.ngmn.org/fileadmin/ngmn/content/images/news/ngmn\_news/NGMN\_5G\_White\_Paper\_V1\_0.pdf
5. ETSI. Multi-access edge computing. http://www.etsi.org/technologies-clusters/technologies/multi-access-computing
6. *Mobile-Edge Computing*. White Paper. Sophia Antipolis, France: ETSI; 2014. https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge\_computing\_-\_introductory\_technical\_white\_paper\_v1%2018-09-14.pdf
7. ETSI. Network functions virtualisation (NFV). https://www.etsi.org/technologies-clusters/technologies/nfv
8. Wikipedia: The Free Encyclopedia. C-RAN. https://en.wikipedia.org/wiki/C-RAN
9. 5G-PPP. 5G-PPP phase 1 projects. https://5g-ppp.eu/5g-ppp-phase-1-projects/
10. 5G-PPP. Global 5G actions. https://5g-ppp.eu/global-5g-actions/
11. Garg VK. *Wireless Network Evolution: 2G to 3G*. 1st ed. Rappaport TS, ed. Upper Saddle River, NJ: Prentice Hall PTR; 2001.
12. Dahlman E, Parkvall S, Sköld J. *4G: LTE/LTE-Advanced for Mobile Broadband*. Waltham, MA: Academic Press; 2013.
13. Amit N, Gordon A, Har'El N, et al. Bare-metal performance for virtual machines with exitless interrupts. *Commun ACM*. 2016;59(1):108-116.
14. Kernel virtual machine (KVM). http://www.linux-kvm.org
15. Kivity A, Kamay Y, Laor D, Lublin U, Liguori A. kvm: the Linux virtual machine monitor. In: Proceedings of the Linux Symposium; 2007; Ottawa, Canada.
16. QEMU. http://www.qemu.org
17. Bartholomew D. QEMU: a multihost, multitarget emulator. *Linux Journal*. 2006;2006(145):3.
18. VMware. http://www.vmware.com
19. Rosenblum M. VMware's virtual platform. In: Proceedings of Hot Chips; 1999; Stanford, CA.
20. Linux Containers (LXC). https://linuxcontainers.org
21. Virtual Box. https://www.virtualbox.org
22. Xen Project. http://www.xenproject.org
23. Alaez RM, Chirivella-Perez E, Calero JMA, Wang Q. New topology management scheme in LTE and 5G networks. Paper presented at: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring); 2018; Porto, Portugal.
24. Sesia S, Toufik I, Baker M. *LTE - The UMTS Long Term Evolution: From Theory to Practice*. Hoboken, NJ: John Wiley & Sons; 2011.
25. ElNashar A, El-Saidny MA, Sherif M. *Design, Deployment and Performance of 4G-LTE Networks: A Practical Approach*. Hoboken, NJ: John Wiley & Sons; 2014.
26. Baldo N. The ns-3 LTE module by the LENA project. 2011.
27. Henderson TR, Lacage M, Riley GF, Dowell C, Kopena J. Network simulations with the ns-3 simulator. Paper presented at: SIGCOMM Demonstration. 2008; Seattle, WA.
28. Nikaein N, Marina MK, Manickam S, Dawson A, Knopp R, Bonnet C. OpenAirInterface: a flexible platform for 5G research. *ACM* *SIGCOMM Comput Commun Rev*. 2014;44(5):33-38.
29. SP-030388 network sharing cover sheet - 3GPP. ftp://www.3gpp.org/tsg\_sa/TSG\_SA/TSGS\_21/Docs/PDF/SP-030388.pdf
30. 3GPPP. http://www.3gpp.org/
31. Network functions virtualisation(NFV); management and orchestration. Sophia Antipolis, France: ETSI; 2014. http://www.etsi.org/ deliver/etsi\_gs/NFV-MAN/001\_099/001/01.01.01\_60/gs\_nfv-man001v010101p.pdf

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | 755 | |  |
|  |  |  |  |
|  | CHIRIVELLA-PEREZ ET AL. |  |  | **15** | |  |
|  |  |  |  |  |  |  |



1. Mijumbi R, Serrat J, Gorricho JL, Latré S, Charalambides M, Lopez D. Management and orchestration challenges in network functions virtualization. *IEEE Commun Mag*. 2016;54(1):98-105.
2. MAAS: Metal As A Service. http://maas.io
3. Rocks. http://www.rocksclusters.org/rocks-documentation/4.1/getting-started.html
4. Hewlett Packard Enterprise Development. HPE insight cluster management utility. http://www8.hp.com/us/en/products/server-software/product-detail.html?oid=3296361
5. Chirivella-Perez E, Alaez RM, Calero JMA, Wang Q, Gutiérrez-Aguado J. UWSIO: towards automatic orchestration for the deployment of

5G monitoring services from bare metal. Paper presented at: 2018 IEEE Wireless Communications and Networking Conference (WCNC); 2018; Barcelona, Spain.

1. OpenEnergyMonitor. https://openenergymonitor.org/emon/
2. Arduino. https://www.arduino.cc/
3. Smart Energy Groups. Energy awareness for everybody. https://smartenergygroups.com/
4. eNOS - A Cloud Based, Open Source Energy Management. http://www.openenos.com/
5. Symantec. Ghost solution suite. https://www.symantec.com/en/uk/ghost-solution-suite/
6. Clonezilla. http://clonezilla.org/
7. PartImage. https://www.partimage.org/Main\_Page
8. Fog Project. https://fogproject.org/
9. VirtualBox. Chapter 5. Virtual storage. https://www.virtualbox.org/manual/ch05.html
10. McLoughlin M. The QCOW2 image format. 2008.
11. Puppet. https://puppetlabs.com/
12. CHEF. https://www.chef.io/chef/
13. Marschall M. *Chef Infrastructure Automation Cookbook*. Birmingham, UK: Packt Publishing Ltd; 2015.
14. SmartFrog. http://smartfrog.org/display/sf/SmartFrog+Home
15. Goldsack P, Guijarro J, Loughran S, et al. The SmartFrog configuration management framework. *ACM SIGOPS Oper Syst Rev*. 2009;43(1):16-25.
16. Juju. https://jujucharms.com/
17. Ansible. https://www.ansible.com/application-deployment
18. Hochstein L, Moser R. *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. Sebastopol, CA: O'Reilly Media Inc; 2017.
19. Chirivella-Perez E, Gutiérrez-Aguado J, Alcaraz-Calero JM, Wang Q. NFVMOn: enabling multioperator flow monitoring in 5G mobile edge computing. *Wirel Commun Mob Comput*. 2018;2018.
20. Wikipedia: The Free Encyclopedia. Cloud computing: infrastructure as a service. https://en.wikipedia.org/wiki/Cloud\_computing# Infrastructure\_as\_a\_service\_.28IaaS.29
21. Bhardwaj S, Jain L, Jain S. Cloud computing: a study of infrastructure as a service (IAAS). *Int J Eng Inf Technol*. 2010;2(1):60-63.
22. Sefraoui O, Aissaoui M, Eleuldj M. OpenStack: toward an open-source solution for cloud computing. *Int J Comput Appl*. 2012;55(3):38-42.
23. Kumar R, Jain K, Maharwal H, Jain N, Dadhich A. Apache cloudstack: open source infrastructure as a service cloud computing platform.

*Int J Adv Eng Technol Manag Appl Sci*. 2014;1(2):111-116.

1. Milojiciˇc´ D, Llorente IM, Montero RS. Opennebula: a cloud management tool. *IEEE Internet Comput*. 2011;15(2):11-14.
2. Medved J, Varga R, Tkacik A, Gray K. Opendaylight: towards a model-driven SDN controller architecture. Paper presented at: 2014 IEEE 15th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM); 2014; Sydney, Australia.
3. Berde P, Gerola M, Hart J, et al. ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking; 2014; Chicago, IL.
4. Ryu S. Framework community: Ryu SDN framework. 2015.
5. Wallner R, Cannistra R. An SDN approach: quality of service using big switch's floodlight open-source controller. *Proc Asia Pac Adv Netw*. 2013;35:14-19.
6. Frantz PJ, Thompson GO, inventors; International License Exchange Of America LLC, assignee. VLAN, frame format. Google Patents. US Patent 5,959,990. 1999.
7. Wikipedia: The Free Encyclopedia. Virtual Extensible LAN. https://en.wikipedia.org/wiki/Virtual\_Extensible\_LAN
8. Mahalingam M, Dutt D, Duda K, et al. RFC 7348: virtual extensible local area network (VXLAN): a framework for overlaying virtualized layer 2 networks over layer 3 networks. 2014.
9. Wikipedia: The Free Encyclopedia. Generic routing encapsulation. https://en.wikipedia.org/wiki/Generic\_Routing\_Encapsulation
10. Farinacci D, Li T, Hanks S, Meyer D, Traina P. Generic routing encapsulation (GRE). RFC No 2784. 2000.
11. Virtual tenant networks (VTN). http://docs.opendaylight.org/en/stable-oxygen/user-guide/virtual-tenant-network-(vtn).html
12. Service function chainning. https://wiki.opnfv.org/display/sfc/Service+Function+Chaining+Home
13. Semeria C. *Multiprotocol Label Switching: Enhancing Routing in the New Public Network*. White Paper. Mountain View, CA: Juniper Networks; 2000.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 756 | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |
| **16** | |  |  | CHIRIVELLA-PEREZ ET AL. | |  |
|  |  |  |  |  |  |  |



1. Wikipedia: The Free Encyclopedia. Multiprotocol label switching. https://en.wikipedia.org/wiki/Multiprotocol\_Label\_Switching
2. Bhamare D, Jain R, Samaka M, Erbad A. A survey on service function chaining. *J Netw Comput Appl*. 2016;75:138-155.
3. Halpern J, Pignataro C. Service function chaining (SFC) architecture. RFC 7665. 2015.
4. OpenStack. Neutron. https://wiki.openstack.org/wiki/Neutron
5. Tkachova O, Salim MJ, Yahya AR. An analysis of SDN-OpenStack integration. Paper presented at: 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T); 2015; Kharkiv, Ukraine.
6. 5G-PPP/Selfnet. Deliverable 2.1: use cases definition and requirements of the system and its components. https://bscw.selfnet-5g.eu/pub/ bscw.cgi/d18783/SELFNET%20Deliverable%202.1%20-%20Final%20v12.pdf
7. Pérez MG, Celdrán AH, Ippoliti F, et al. Dynamic reconfiguration in 5G mobile networks to proactively detect and mitigate botnets. *IEEE* *Internet Comput*. 2017;21(5):28-36.
8. Martínez R, Vilalta R, Requena M, Casellas R, Muñoz R, Mangues J. Experimental SDN control solutions for automatic operations and management of 5G services in a fixed mobile converged packet-optical network. Paper presented at: 2018 International Conference on Optical Network Design and Modeling (ONDM); 2018; Dublin, Ireland.
9. Katsalis K, Nikaein N, Schiller E, Ksentini A, Braun T. Network slices toward 5G communications: slicing the LTE network. *IEEE Commun* *Mag*. 2017;55(8):146-154.

**How to cite this article:** Chirivella-Perez E, Marco-Alaez R, Hita A, et al. SELFNET 5G mobile edge computing

i**n**frastructure:Designandandprototyping.Softw:.*Softw:*Pract*Pract*Exper*Exper*.2020;50:741.2019;1–16–756.[https://doi](https://doi.org/10.1002/spe.2681).[https://doi.org/10.org/10.10.1002/spe2/spe.2681.2681](https://doi.org/10.1002/spe.2681)