Optimal Service Placement with QoS Monitoring in NFV and Slicing Enabled 5G IoT Networks

Andra-Isabela-Elena Ciobanu*, Cosmin Contu*, Eugen Borcoci*, Marius-Constantin Vochin*, and Frank Y. Li§
*Doctoral School of ETTI, University Politehnica of Bucharest, Bucharest, Romania
§Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Email: andraciobanu90@yahoo.com; {cosmin.contu; eugen.borcoci; marius.vochin}@elcom.pub.ro; frank.li@uia.no

***Abstract*—Network function virtualization (NFV) and network slicing are two promising enabling technologies for 5G networks. Considering the volume of data traffic generated by Internet of things (IoT) applications and their service requirement diversity as well as that network resources are spread across different locations, it is imperative to find solutions for optimal service placement and resource allocation for quality of service (QoS) provisioning. In this paper, we address the challenges of optimal network service placement with active QoS monitoring in NFV and network slicing enabled 5G IoT networks and propose a network architecture with optimal computation and resource placement over core, local, and edge data centers. The solution is implemented through virtualized infrastructure managers where operation costs and QoS requirements are considered for service placement. Optimal algorithms are developed based on a control system hub platform with an open source management and orchestration framework. To monitor the performance during traffic runtime, virtual charmed factors are adopted for control and QoS measurement.**

I. Introduction

Today, vertical applications ranging from smart city, smart home to Industry 4.0 are just a few examples that technologies such as 5th generation (5G) mobile networks and the Internet of things (IoT) are reshaping our society towards a super-smart society paradigm. Among various technologies for facilitating 5G and IoT services, network function virtualization (NFV) and network slicing are two prominent enablers as they support the development of flexible and customizable virtual networks for diverse services, especially in multi-tenant and multi-domain environments. To meet the stringent quality of service (QoS) requirements for 5G applications, these two technologies need to be operated in a coordinated manner based on actual network infrastructures and service requirements.

Existing work towards this direction has been focused on finding solutions on how to monitor different QoS parameters (including bandwidth, latency, packet error rate, etc.) and to assure service quality considering the constraints of virtualized network functions (VNFs) and network resources [1]. With respect to network resource allocation, efforts have been made on how to manage workloads and costs in an optimal way given the existence of various types of data centers (DCs) nowadays [2]. Indeed, network resources for today's networks are often spread across core, regional, and local DCs, edge computing, as well as on client premises. To interconnect these DCs and coordinate services, another virtual network entity, virtualized infrastructure manager (VIM), is introduced.

As network infrastructures are becoming more complex and IoT traffic is generally dynamic, it is a challenging task to decide manually where resources should be located in and to allocate them adaptively according to service requirements. To address these challenges, joint considerations of NFV and network slicing resources together with QoS requirements are beneficial. In NFV, various VNF resources are managed through a management and orchestration (MANO) architectural framework. For optimal service placement and resource management, the control system hub (CSH) platform with open source MANO (OSM) provides a promising tool.

In this paper, we first discuss the challenges of optimal network services placement, and active QoS monitoring in 5G networks. Then we propose a solution framework addressing cost and service placement analysis and scalability issues for optimal QoS provisioning. The solution is developed based on our proposed architecture (which includes CSH, VIMs and VNFs) and an existing orchestration NFV platform (OSM). The optimal placement algorithm is implemented based on the CSH network analytics platform along with virtual charmed factors for control and QoS measurements and monitoring. Furthermore, computation and placement of VNFs over relevant VIMs by matching the network specific requirements to the infrastructure availability, runtime metrics, and network costs (self-optimization, or optimization with latency constraint and with scalability constraint) are accomplished and experiments are performed.

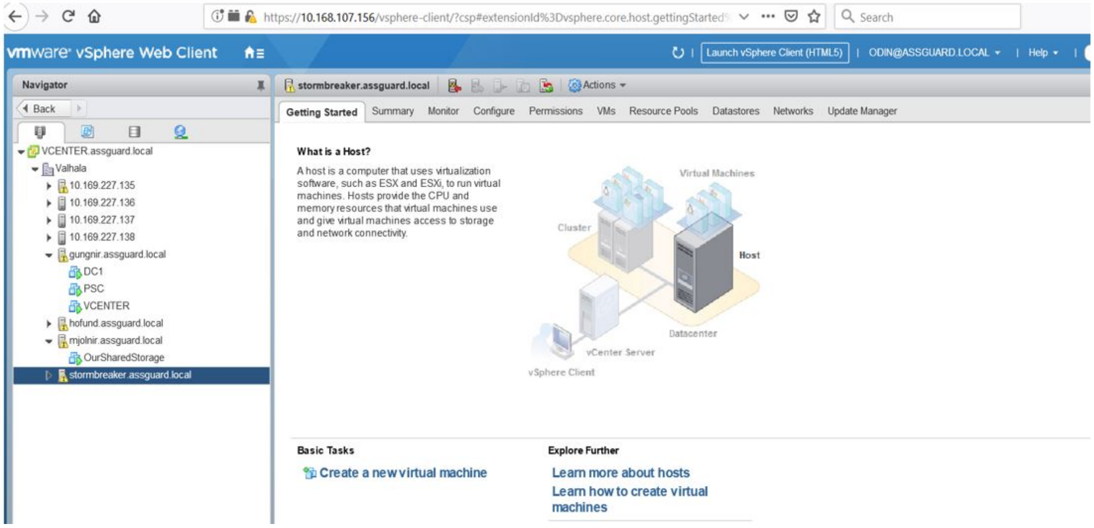
The rest of this paper is organized as follows. Sec. II provides preliminaries and introduces tools used in our implementation. Then we present problem statement and solution design in Sec. III and implementation architecture in Sec. IV. Furthermore in Sec. V, we describe our implementation and experiments. Finally, the paper is concluded in Sec. VI.

II. Preliminaries and Implementation Tools

In this section, we first provide preliminaries to NFV and network slicing. Then a few tools are presented.

A. Network Function Virtualization and Network Slicing

NFV is an emerging powerful architecture based mostly on the concept of virtualization. NFV enables the deployment of



https://10.168.107.156/vsphere-client/?csp#extensionId%3Dvsphere.core.host.gettingStarted

What is a Host?

A host is a computer that uses virtualization software, such as ESX and ESXi, to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity.

Fig. 1: Illustration of the VIM VMware integrated with OpenStack adopted in this study.

originally hardware based proprietary network functions on virtual environments, leveraging the cost efficiency and time-to-market benefits of cloud computing [3]. VNFs are deployed on virtual machines (VMs), which can be chained together in a co-located or distributed cloud environment, offering value-added networks or services [4].

The NFV architectural framework [5] consists of the following components. 1) VNFs that are software implementations of network functions deployed on virtual environments; 2) NFV infrastructure (NFVI) that comprises the logical environment's building blocks, i.e., storage, computing, network, and their respective assisting hardware components; 3) MANO that is responsible for managing and orchestrating VNFs and the NFVI; and 4) VIM that controls and manages the NFVI associated resources that usually belong to a single network operator. Depending on its setup, a VIM may be dedicated, controlling a specific type of NFVI resource, e.g., a computing resource, or for managing multiple NFVI resources.

Network slicing is a novel concept that optimizes resource allocation, decreases operational costs, and at the same time increases energy efficiency. It allows operators to partition their networks into dedicated slices for service delivery using a portion of their network resources for a specific customer or service [6]. The concept of network slicing provides a promising solution for many use cases in 5G such as IoT, Internet to vehicles (I2V), smart energy grid, and smart homes.

B. Tools and Components Used in Our Implementation

1) OSM: As an ETSI-hosted open source community, OSM delivers a production quality MANO stack for NFV. OSM can handle common and standardized information models and it is suitable for all VNFs including both operationally significant and VIM independent [8]. OSM is aligned to the NFV industry specification group information models and it provides first-hand feedback based on its implementation experience. Accordingly, OSM has been adopted in this study to create VNFs, network services, and slices, as well as to instantiate them.

2) VIM-OpenStack and VMware: OpenStack is a cloud operating system that controls large pools of computing, storage, and networking resources throughout a DC [9]. All OpenStack operations are managed and provisioned through application programming interfaces (APIs) with common authentication mechanisms.

For the implementation performed in this study, OpenStack has been adopted as the tool for implementing the cloud computing part of the architecture (to be presented in Sec. IV), as well as for the VIM part. More specifically, the selected hypervisor is VMware ESXi (which is compatible with the OpenStack Nova Compute) with an LSI Drive Controller (SATA/SAS-MegaRAID SAS 936) and it has 2 controllers for queue depth length. Another important element in our implementation is the physical network which consists of a configurable physical switch and a gateway firewall appliance. The gateway appliance, Vyatta Brocade, acts as a router and as a firewall in order to take control of all physical traffic inside a DC and between both DCs (OSM and CSH). The virtual network has been realized with the NSX module where a standard switch feature in vCenter has been selected. Fig. 1 illustrates the adopted VIM VMware in this study which is integrated with OpenStack .

3) CSH Software as a Service (SaaS): A CSH platform contains typically three servers, i.e., the Data, Core, and Analytics components. These components are hosted on separate servers to provide higher capacity and QoS that a production environment needs [10].

As the Core component of our architecture, the main part of the CSH platform consists of an IBM WebSphere Application Server ISH (Service Hub) which contains the software developed to automate the VNFs and slices for optimal allocation to VIMs, based on QoS constraints and operation costs. It has also the ability to attach a VNF to a specific DC automatically and to assure connectivity to a specific region or location. So, there are some sort of anchors to consider when a network service (NS) is deployed and then CSH gives that information

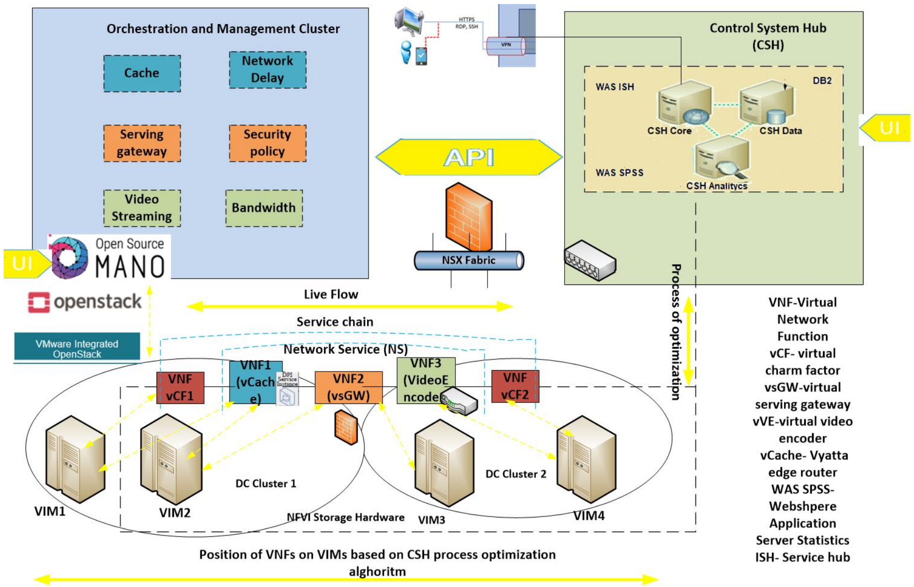


Fig. 2: Illustration of multi-VIM network slicing with QoS assurance.

to OSM (through APIs). Afterwards, the NS might be a part of some bigger external services which may not be visible to OSM. A CSH platform can be hosted in the cloud or on premises. Fig. 2 illustrates the composition of a CSH as well as its relationships with OSM and OpenStack.

CSH Analytics is the place where traffic for virtual charmed factors (vCFs) are generated. A vCF is basically an active VNF and it acts like a virtual test agent. It can generate traffic. It can also do real service requests in order to measure through the entire service chain and to make sure it works. A vCF can be also attached to an existing VNF through a software code. For in-depth data exploration, reporting, and modeling, an IBM statistical package for the social sciences (SPSS) Analytics client software platform has been adopted.

In CSH Data, there is information stored about the templates of vCFs. This server also stores the reports of QoS monitoring which can be sent to end-users. As shown in Fig. 2, CSH communicates with OSM through an API. It also coordinates the operation of vCFs.

4) Charms: A charm is a piece of software that runs scripts over some targets. Traditionally, the charms written in Juju are used inside an application or in the same machine as an application. Juju is an open source modeling tool composed of a controller, models and charms, for operation software in the cloud [11].

III. Problem Statement and Solution Design

To develop an NFV and network slicing based solution for 5G IoT networks with QoS constraint and cost optimization in mind, we need to explore a few aspects that are essential for our architecture design and implementation (which will be presented in the next two sections), as presented below.

A. Cost Models and Service Position Analytics

1) Service Position Analytics: 5G infrastructures are expected to provide QoS and assure requested quality of experience, supporting flexible business models, use cases, and applications. End-users can negotiate with their service providers on different QoS levels to meet their expectations [12].

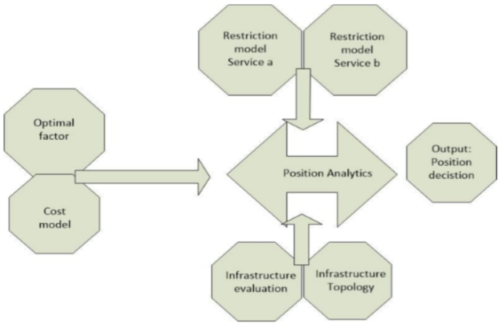

Restriction model Service a
Restriction model Service b
Optimal factor
Cost model
Position Analytics
Output: Position decistion
Infrastructure evaluation
Infrastructure Topology

Fig. 3: Overview of position analytics: Factors and procedures [13].

As a service can be provided from various locations at different costs, it is important to find out optimal locations for service provisioning while keeping QoS requirements satisfied. An optimal position of a service does not only rely on cost and computation but also depends on the relationships between network operators and customers, as well as between data sources and destinations. In order to find an optimal position for a service, different factors need to be taken into consideration, such as possible locations, operation costs, and service efficiency. Fig. 3 illustrates the relationship and procedure for service position analytics in cloud and DC networks.

2) Service Placement Principles: The placement function and algorithm from CSH shall consider all VIMs that are available to end-users. A playground function will make sure that all the constraints, e.g., latency or bandwidth, are met at an optimal cost. By cost optimization, the cheapest way to deploy a service over the available set of infrastructure can be found. The following principles are considered in this study.

- To provide a network service without any QoS constraint, different costs may apply when the service is deployed in other DCs. In this case, the placement for deployment

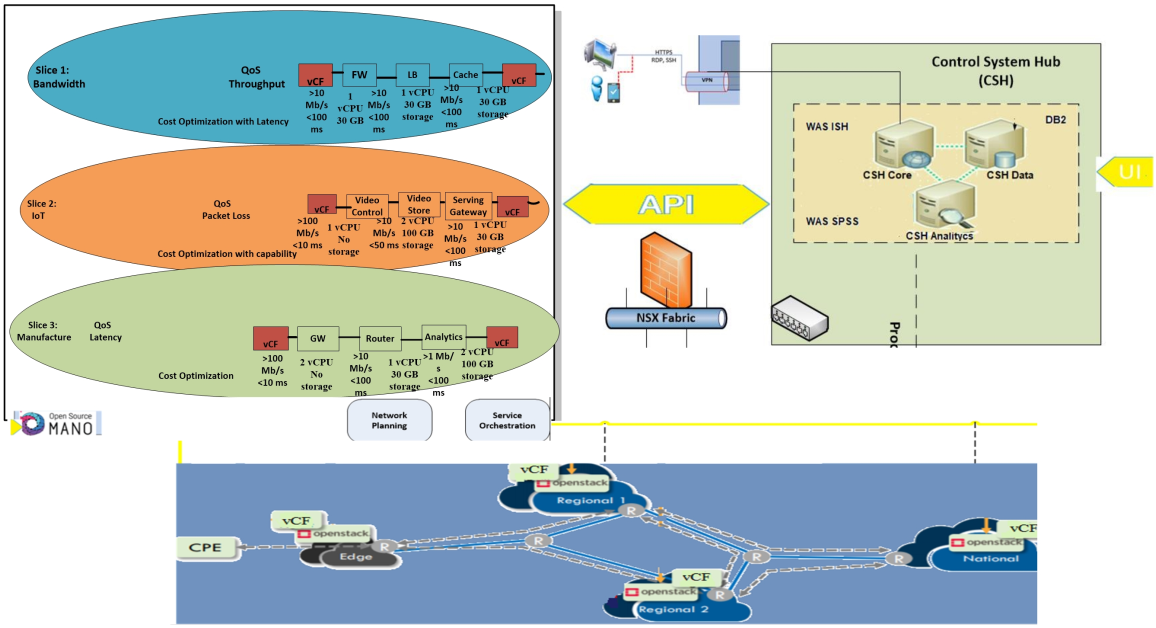


Fig. 4: Implementation of three network slices based on OSM, CSH, and OpenStack. Four DCs are configured in the NFV architecture.

will be solely based on costs.

- For cost optimization with QoS constraints, a decision is made based on where this network service can be distributed with the lowest operation cost. Fig. 4 depicts an architecture example where cost optimization for three scenarios (which will be presented in the next section) is considered.

B. NFV and Network Slicing for IoT Services

Considering the heterogeneity of IoT traffic, how to deal with dynamic network resource allocation across multiple DCs and to perform cross-platform behavior optimization need to be addressed [14]. To provide IoT services based on the concepts of NFV and slicing, new solutions for autonomic administration to make network management *proactive* rather than *reactive* need to be developed.

As an effort of this study, we develop a software python coded charmed VNF, called vCF, which is responsible to monitor different specified parameters of a VNF during traffic runtime, in a *proactive* manner. Based on this vCF, different states/contexts in the VNFs, NSs or service chains, workloads, QoS and security policies or other parameters can be observed and adjusted during the operation time, without notifying end-users or customers directly.

C. Scalability and Interoperability

In IoT networks where a large number of devices may require simultaneous connectivity, there are two types of scalability. While vertical scalability addresses the addition or removal of an IoT node's computing resources, horizontal scalability deals with the dimension of a network with respect to number of computing resources etc. [15]. On the other hand, every IoT network solution contains a mixture of various components and systems. Due to this diversity, interoperability is a must for IoT services.

To address these issues, our design considers connections among VIMs and their interconnecting links. For example, after identifying the end points between two different VIMs, we consider the cost to connect them as well as the QoS constraint. This solution can avoid fragmentation of network resources and lower down the integration costs of an IoT solution in the long term by building an interoperable architecture from the start.

IV. Network Slices and NFV Architecture

In this section, we first present three network scenarios and formulate the corresponding cost optimization problem for each scenario. Then a high-level VNF architecture including three network slices are outlined, each corresponding to one of these three scenarios.

A. Network Scenarios and Slices

Three network slices based on three scenarios are envisaged in this study as presented below. For each scenario, we consider that a network slice is allocated to a certain type of customers or a service. A cost optimization problem is defined for each scenario.

- **Scenario/Slice 1:** A dedicated slice for 5G residential subscribers for bandwidth ensured services in terms of throughput. The slice is enabled with QoS measurement, vCache, firewall, and load balanced VNFs. In this case, the task of the cost optimization algorithm is to reduce latency between the national DC and a local DC (VIM1 and VIM2 respectively as shown in Fig. 4, with throughput as a constraint.
- **Scenario/Slice 2:** An IoT/IoV slice which demands low packet loss as a QoS requirement for video transcode. The slice uses VNF Fortinet as a security gateway (sGW) which can be considered as a radio access network (RAN). In this case, cost optimization with capability

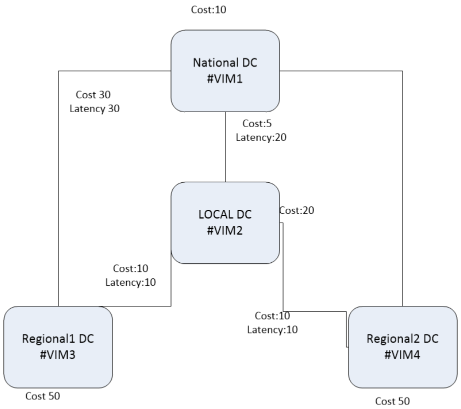


Fig. 5: Configuration of costs and QoS constraints for experiments.

applies to minimal packet loss between two regional DCs (VIM3 and VIM4 respectively as shown in Fig. 4, with packet loss as a constraint. The other VNFs adopted in this slice are video control and video store where the latter one is considered for an IoV case.

- **Scenario/Slice 3:** A manufacture slice with low latency as the QoS requirement. In this case, cost optimization is meant for the VNF as a router for performing self-cost optimization analysis, with latency as a constraint.

B. High-Level Architecture

From a high-level perspective, the proposed architecture consists of three components, as shown in Fig. 2 and explained below. A more detailed architecture including the three slices defined above is illustrated Fig. 4.

- OSM is the component that performs management orchestration considering various aspects from QoS requirements, serving gateway placement, to security policy. It also mandates the deployment of VNFs, network services, service chain and slices, as well as test agents.
- CSH is the platform that is responsible for the coordination among vCFs and it contains a CSH cache, database, and analytics. It also offers the automation workflows and has a database containing the costs of links and the workload distribution of VNFs among VIMs. CSH interacts with the orchestrator, OSM, through an API.
- OpenStack partitions VNFs on VIMs based on a CSH process optimization algorithm. It is used as VIMs building up inter-operation among VNF blocks.

V. Implementation and Experiments

Based on the architecture and platform described above, we have implemented three network slices which are presented in Sec. IV and shown in Fig. 4. In this section, we summarize our implementation and present our experiments.

A. Network Configuration and Implementation Brief

Consider an NFV enabled network where the VIM infrastructure is composed of four DCs (including one local/edge DC, two regional DCs, and one national DC). The customer premises equipment (CPE) is directly connected to the local DC and can be connected to one of the two regional DCs or the national DC. The network topology of our implementation is illustrated in the lower part of Fig. 4 and the links between any DCs along with the configured costs and latency are shown in Fig. 5. Inside each DC, there is a VIM. vCF agents are also attached in each DC to capture active latency metrics between any DCs.

To enable cost optimization in this network, we need to configure the costs to deploy a VNF to a specific VIM and the costs to use the links between different VIMs. For our implementation, two different configuration files have been created. One is called “vnf_charge_list.yaml” and the other one is named “lip_charge_list”. While the first file records the interconnection links between any points of presence, the second file gives a price list for the interconnection links.

These configuration files are copied into the release using docker commands (docker swarm). For each identified container, these files need to be copied into the container at the specific location.

B. Network Slice Demonstration and Cost Optimization

To demonstrate network slicing in the implemented NFV architecture, three layouts representing three different types of slices have been developed, as illustrated in the upper-left part of Fig. 4. To enable QoS measurements and cost optimization, various functions are implemented in the CSH based on the network topology presented above. In Fig. 5, the costs and QoS constraints in terms of latency are labeled along the links.

Slice 1: For the bandwidth dedicated slice, we pin VNF1 (marked as FW in Fig. 4) to VIM4. When considering the latency of the link between VNF2 and VNF3 (LB and Cache in Fig. 4 respectively), we conclude that this link cannot be selected because it does not satisfy the latency requirement.

Alternatively, we may deploy all functions in VIM4, but that is not so efficient. Instead, the most cost-efficient way would be to place VNF1 in VIM4 since VNF2 and VNF3 are free of charge. Then they would go to the national DC VIM1 first because this option satisfies the latency requirement and operations would be kept as cheap as possible.

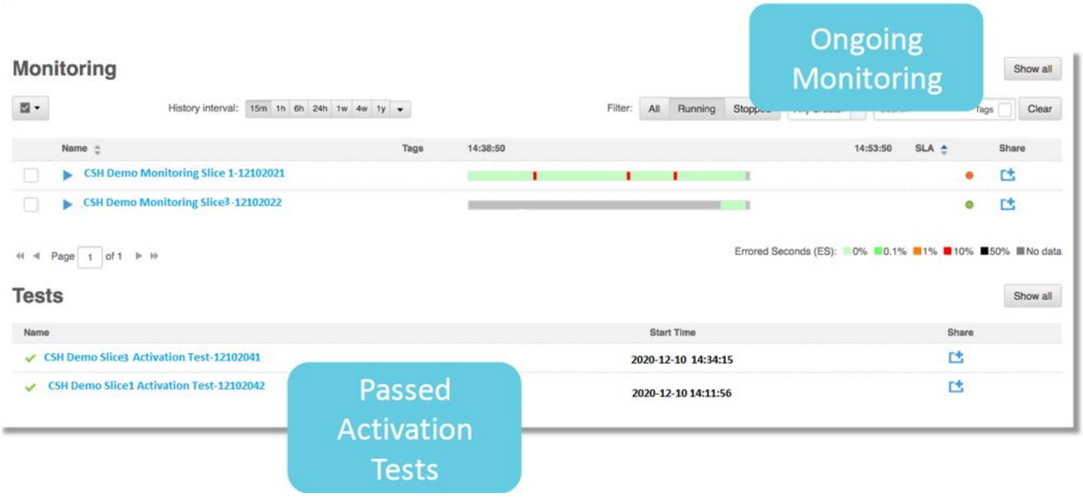
Slice 2: For the IoT slice, an sGW VNF will be placed at VIM3, and the other servers (for video control/video storage) will be placed according to the result of cost optimization. Consequently, VNF placement will be distributed (one VNF in the local DC (VIM2), one in the national DC (VIM1), and another one in the regional DC (VIM3)).

In this way, we achieve the most efficient cost deployment. This example reveals the benefit of automatic service position supported by cost optimization.

Slice 3: For the manufacture slice, no specific QoS constraint is defined. Following the principle presented in Subsec. III-A, the VNF allocation will end up with the service position which leads to the cheapest cost, i.e., the local DC (VIM2) in this example.

C. Test Activation and QoS Monitoring

Based on the implementation presented above, we can perform experiments by activating corresponding VNF modules


Ongoing Monitoring
Passed Activation Tests
Show all
History interval: 15m 1h 6h 24h 1w 4w 1y
Filter: All Running Stopped
Tags
Clear
Page 1 of 1
Errored Seconds (ES): 0% 0.1% 1% 10% 50% No data
Show all

Monitoring

Name	Tags	14:38:50	14:53:50	SLA	Share
CSH Demo Monitoring Slice 1-12102021				●	
CSH Demo Monitoring Slice3 -12102022				●	

Tests

Name	Start Time	Share
✓ CSH Demo Slice3 Activation Test-12102041	2020-12-10 14:34:15	
✓ CSH Demo Slice1 Activation Test-12102042	2020-12-10 14:11:56	

Fig. 6: Illustration of implemented QoS monitoring and experiments.

based on a selected slice and services. The steps of the entire workflow are summarized as follows:

a) Blueprint/Pattern items: VNFDescriptors (VNFDs) and Descriptors. First, it starts with the elaboration of VNFDs in the OSM user interface, then more specifically in the OSM catalog.
b) Blueprint activation: Test the templates for three different constraint models presented in this study. This part is achieved in the CSH Hub where specific parameters and maps are defined.
c) Instantiation of the implemented VNFs and services.
d) Start the VNFs and test agents vCFs.
e) Capture network properties.

With the support of vCF agents and through the analytic component of the CSH, balanced workloads among different VIMs are achieved and satisfied QoS to end-users is ensured. Fig. 6 is a screen shot of one of our experiments, illustrating the proper operation of the implemented system for service provisioning and active monitoring.

VI. Conclusions

The paper addressed an important task in NFV and network slicing enabled 5G networks on VNF resource placement considering the constraints of QoS requirements. A solution based on our proposed architecture example (CSH) and an existing orchestration NFV platform (OSM) has been developed. The optimal placement algorithm employs the CSH network analytics platform along with virtual charmed factors for control and QoS measurements and monitoring. Based on an implemented network topology with four data centers and several VIMs and VNFs, optimal placements of VNFs over associated VIMs are assessed and experiments are performed.

VII. Acknowledgment

The research leading to these results has received funding from the NO Grants 2014-2021, under project contract no. 42/2021, RO-NO-2019-0499 - “A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services (SOLID-B5G)”.

References

[1] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. G. Martini, and L. Atzori, “QoE management of multimedia streaming services in future networks: A tutorial and survey,” *IEEE Commun. Surveys & Tuts.*, vol. 22, no. 1, pp. 526–565, 1st Quart., 2020.

[2] F. P. Tso, S. Jouet, and D. P.Pezaros, “Network and server resource management strategies for data centre infrastructures: A survey,” *Comput. Netw.*, vol. 106, pp. 209–225, Sep. 2016.

[3] ETSI, “Network functions virtualization: An introduction, benefits, enablers, challenges & call for action,” NFV White Paper, Issue 1, Oct. 2012.

[4] J. de. J. Gil Herrera and J. F. Botero Vega, “Network functions virtualization: A survey,” *IEEE Latin America Trans.*, vol. 14, no. 2, pp. 983–997, Mar. 2016.

[5] ETSI, “Network functions virtualization (NFV); Architectural framework,” ETSI GS NFV 002, v1.2.1, Dec. 2014.

[6] J. Menglan, C. Massimo, and T. Mahmoodi, “Network slicing management & prioritization in 5G mobile systems,” in *Proc. European Wireless*, May 2016, pp. 1–6.

[7] K. M. Alam, M. Saini, and A. El Saddik, “tNote: A social network of vehicles under Internet of things,” Book Chapter in *Internet of Vehicles–Technologies and Services*, Berlin, Germany: Springer-Verlag, pp. 227–236, 2014.

[8] Open Source MANO, “OSM Release FIVE documentation,” 2019. [Online]. Available: https://osm.etsi.org/wikipub/ index.php/OSM_Release_FIVE_Documentation.

[9] C. Campolo, A. Molinaro, A. Iera, R. R. Fontes, and C. E. Rothenberg, “Towards 5G network slicing for the V2X ecosystem,” in *Proc. IEEE NetSoft*, Jun. 2018, pp. 400–405.

[10] A. Ciobanu, E. Borcoci, and M. Vochin, “A quality-of-service scenario awareness for use-cases of open-source management and control system hub in edge computing,” in *Proc. IEEE BlackSea*, May 2021, pp. 1–5.

[11] I. Afolabi, A. Ksentini, M. Bagaa, T. Taleb, M. Corici, and A. Nakao, “Towards 5G network slicing over multiple-domains,” *IEICE Trans. Commun.*, vol. E110.B, no. 11, pp. 1992–2006, Nov. 2017.

[12] 5G NORMA, “Definition of connectivity and QoE/QoS management mechanisms - intermediate report,” Project Deliverable 5.1, v1.0, Nov. 2016.

[13] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” *SIGCOMM Comput. Commun. Review*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[14] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, “A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[15] M. A. Imran, A. Zoha, L. Zhang, and Q. H. Abbasi, “Grand challenges in IoT and sensor networks,” *Front. Comms. Net.*, vol. 1, article 619452, pp. 1–6, Dec. 2020.