The Path Towards Resource Elasticity for 5G Network Architecture

David M. Gutierrez-Estevez*, Marco Gramaglia†, Antonio de Domenico‡, Nicola di Pietro‡,
Sina Khatibi§, Kunjan Shah§, Dimitris Tsolkas¶, Paul Arnold‖, Pablo Serrano†
*Samsung Electronics R&D Institute UK, † Universidad Carlos III of Madrid, Spain ‡ CEA-LETI, MINATEC, France
§ Nomor Research GmbH, Germany ¶ University of Athens, Greece ‖ Deutsche Telekom AG, Germany

***Abstract*—Vertical markets and industries are addressing a large diversity of heterogeneous services, use cases, and applications in 5G. It is currently common understanding that for networks to be able able to satisfy those needs, a flexible, adaptable, and programmable architecture based on network slicing is required. Moreover, a softwarization and cloudification of the communications networks is already happening, where network functions (NFs) are transformed from monolithic pieces of equipment to programs running over a shared pool of computational and communication resources. However, this novel architecture paradigm requires new solutions to exploit its inherent flexibility. In this paper, we introduce the concept of *resource elasticity* as a key means to make an efficient use of the computational resources in 5G systems. Besides establishing a definition as well as a set of requirements and key performance indicators (KPIs), we propose mechanisms for the exploitation of elasticity in three different dimensions, namely *computational elasticity* in the design and scaling of NFs, *orchestration-driven elasticity* by flexible placement of NFs, and *slice-aware elasticity* via cross-slice resource provisioning mechanisms. Finally, we provide a succinct analysis of the architectural components that need to be enhanced to incorporate elasticity principles.**

I. Introduction

The 5th generation (5G) of cellular systems will change the access to technology for users, vertical markets and industries. Thanks to the 5G-enabled technical capabilities, they will experience a drastic transformation that will trigger the development of cost-effective new products and services. A large number of uses cases and corresponding requirements for representative vertical markets such as automotive, health, factories of the future, energy, and media and entertainment will need agile access to network support functionalities [1]. This will require a fundamental rethinking of the mobile network architecture and interfaces. The expected diversity of services, use cases, and applications in 5G requires a flexible, adaptable, and programmable architecture. To this end, network architecture must shift from the current *network of entities* to a *network of capabilities*.

In the context of 5G network architecture, a few key concepts have been introduced in the last years by Standards Development Organizations (SDOs) and research efforts. The first one is the concept of network slicing [2], which allows the network to run multiple network instances in parallel. It was introduced as an effective way to meet all of the heterogeneous requirements from supported use cases and services by means of a cost-effective multi-tenant shared network infrastructure. Another fundamental enabler that emerged as an initiative from the industry to increase the deployment flexibility and the agility with which a new service is integrated within the network is network function virtualization (NFV) and its management and orchestration (MANO) architecture [3]. NFV is a framework where network functions (NFs) that traditionally used dedicated hardware are now implemented in software that runs on top of general purpose hardware, effectively enabling a hardware-software separation that reduces both capital and operational expenditures (*i.e.*, CAPEX and OPEX).

In this paper, we focus on an architectural concept for 5G network architecture that we believe will be key given the above well-established innovations. We refer to this concept as *resource elasticity*. Elasticity is a well-studied concept in cloud computing systems defined as the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible [4], [5]. In networks, temporal and spatial traffic fluctuations require that the network efficiently scales resources such that, in case of peak demands, the network adapts its operation and re-distributes available resources as needed, gracefully scaling the network operation. We refer to this flexibility, which could be applied both to computational and communications resources, as resource elasticity. Although elasticity in networks has already been exploited traditionally in the context of communications resources (e.g., where the network gracefully downgrades the quality for all users if communications resources such as spectrum are insufficient), in this paper we focus on the *computational aspects* of resource elasticity, as we identify the management of computational resources in networks a key challenge of future virtualized and cloudified 5G systems.

The reminder of this paper, dedicated to describe in depth the concept of resource elasticity, is organized as follows. Section II presents a definition of elasticity, along with the main associated requirements and key performance indicators (KPIs). In Section III we cover the main challenges and envisioned mechanisms for provisioning resource elasticity. Section IV shows the architectural components involved in resource elasticity. Finally, conclusions are drawn in Section V.

II. Definition, Requirements and KPIs

As previously discussed, while the concept of elasticity has extensively been addressed in the context of traditional cellular networks, its scope has mainly captured the communications aspects. For instance, network protocols and algorithms have been designed to gracefully deal with shortages in the available bandwidth, with increases in the latency of the communication link, or with a lack of available dedicated antennas. The lack of consideration of computational aspects is due to the relatively novel trend of softwarization and cloudification of networks, now considered key for 5G system architecture. In this context, virtual network functions (VNFs) do not only use communication resources such as the ones previously mentioned, but also those native to the cloud environment, *i.e.*, computational resources such as CPU or memory. Therefore, elasticity should now be enforced in the network in a holistic manner. We next provide a definition of elasticity in this new context, a description of the elastic operation requirements, and a set of KPIs.

A. Resource Elasticity: A Definition

The resource elasticity of a communications system can be defined as *the ability to gracefully adapt to load changes in an automatic manner such that at each point in time the available resources match the demand as closely and efficiently as possible*. Hence, elasticity is intimately related to the system response when changes occur in the amount of available resources. We employ the term gracefully in the definition of elasticity to imply that, for a relatively small variation in the amount of resources available, the operation of the service should not be disrupted. If the service produces a quantifiable output, and the resource(s) consumed are also quantifiable, then the *gracefulness* of a service can be defined as the continuity of the function mapping the resources to the output; sufficiently small changes in the input should result in arbitrarily small changes in the output (in a given domain) until a resource shortage threshold is met where the performance cannot keep up. We refer to this resource shortage threshold as *minimum footprint*. Fig. 1 shows a conceptual example of the operation of an elastic system compared to a non-elastic one, where the elastic performance is capable of achieving graceful degradation with resource shortages until the minimum footprint is met. An elastic VNF should thus be able to cope with variations in the availability of resources without causing an abrupt degradation in the outputs provided by the function.

B. Elastic Operation Requirements

Resource elasticity can be exploited from different perspectives, each of them being a fundamental piece required to bring overall elasticity to the network operation. In this subsection, we describe in detail these different perspectives (referred to as *elasticity dimensions*), which, in turn, generate several innovation opportunities, as we show in Section III.

The first requirement for an elastic network operation is the need for *elasticity at the VNF level*. In general, the concept of elasticity for a NF has not been directly applicable to legacy physical network functions (PNFs). Especially for the case of distributed NFs, the functionality is provided by a physical box that is the result of a thorough joint hardware/software design. Therefore, they have traditionally been designed without any major constraint on the available execution resources as they were expected to be always available by design. In addition, in networks with centralized VNFs, the joint hardware/software design is not possible anymore: VNFs are pieces of software that run on virtual containers on heterogeneous cloud platforms with standard interfaces. Therefore, in this new but already widely adopted scenario, expecting all the needed resources to be always available by design is not reasonable anymore. Furthermore, current VNFs (especially those in the radio access network (RAN)) have been designed under the assumption that required computational resources are always available and they may not be prepared for a shortage of computational resources. Indeed, when such resource outages occur (e.g., lack of CPU availability), current virtualized RAN implementations such as Open Air Interface just drop the frame being processed, and as a result they see their performance severely degraded [6]. This requirement is addressed by the *computational elasticity* innovation area described in Section III-A.

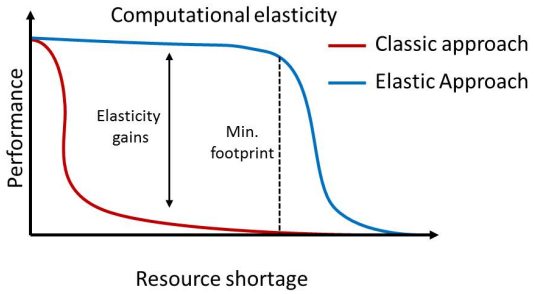

Fig. 1. Illustration of gains achieved by elastic computation.

A second requirement for elastic network operation can be characterized as *elasticity at intra-slice level*. The elastic design of a VNF has an impact on the elasticity of a network slice, defined as the chain of VNFs that provide a telecommunication service. Indeed, chaining and orchestrating a sequence of VNFs with different elastic KPIs (as described in Section II-C) will result in an overall elasticity associated to a tenant running a service using a single network slice. This ultimately affects the quality of experience (QoE) and quality of service (QoS) perceived by users, who may experience different performance degradations according to the elasticity level provided by the tenant. This fact has an impact on the orchestration of a hierarchical cloud architecture such as the one defined in [7], in which the mobile network stack is decomposed into atomic VNFs to better exploit the location diversity and provide service-tailored orchestration. That is, orchestration algorithms may locate VNFs with strong elasticity characteristics where the operational cost is higher or avoid the co-location of inelastic VNFs in the same infrastructure. This

requirement is addressed by the *orchestration-driven elasticity* innovation area described in Section III-B.

The last requirement for elastic operation is *elasticity at the infrastructure level*, *i.e.*, a requirement that involves the infrastructure on which elastic VNFs run. The choice of how many network slices are hosted in the same infrastructure depends on the infrastructure provider who run *e.g.*, admission control algorithms to guarantee that the service level agreement (SLA) with the various tenants are always fulfilled. Elasticity at the infrastructure level is a metric that involves both business and technical KPIs. By leveraging multiplexing gains, more network slices can be hosted on the same infrastructure (thus providing higher revenues), but it comes at the cost of having to resort to more elastic VNFs. This requirement is addressed by the *slice-aware elasticity* innovation area described in Section III-C.

C. Measuring Elasticity: The KPIs

Besides introducing promising opportunities for an optimized operation of the network, the novel concept of resource elasticity also introduces the need of quantifying such gains. These novel elasticity KPIs in some cases may be just mutated from the traditional definitions provided by major SDOs such as 3GPP or ETSI, but some of them are native to this new framework.

A first category of KPIs includes metrics already established such as the service creation time or the availability, the latter defined as the relative amount of time that the function under study produces the output that it would have produced under ideal conditions [8]. A second category, however, includes brand new KPIs that shall be defined to measure the advantages introduced by the elastic operation of the network and the elasticity level of each VNF [9]. Native elasticity KPIs measure mostly the resource savings achieved by the elastic operation, defined as the average cost of deploying and operating the network infrastructure to support the foreseen services. An elastic system should also be able to be optimally dimensioned such that less resources are required to support the same services; furthermore, in lightly loaded scenarios the elastic system should avoid the usage of unnecessary resources and reduce the energy consumption by *e.g.*, consolidating the load, hence also limiting the OPEX.

In addition, another important native metric introduced by elastic VNFs is related to the time component. When a resource shortage occurs, scaling virtual machines or containers that are executing the VNFs is the most likely solution to be adopted. Still, re-orchestration processes usually operate at larger time scales (*i.e.*, seconds), which may not be sufficient for certain services. Even with a graceful resource degradation, the overall QoE metrics may not be fulfilled. This property induces a VNF classification (and the slices using them) according to the capacity of providing graceful performance for a certain time interval before new resources come in. This metric should hence measure how "fragile" a VNF is with respect to the orchestration process, *i.e.*, for how long an elastic function can maintain the KPIs before incurring into an SLA violation, and the kind and amount of resources needed for the VNF to be rescued. We call this KPI *rescuability*: If a VNF can maintain acceptable levels for a very short time and needs a large amount of resources to restore the previous SLA, then it has low rescuability. Conversely, if a VNF can maintain an acceptable level for a long time and need few resources to re-gain normal operation, then it has a high rescuabilty.

Finally, we also envision elasticity-related business-driven KPIs such as the price of resource overbooking, the average performance loss of a single slice in comparison with the monetary gains that additional network slices may provide, or the specific amount of network slices that can be hosted given a total amount of infrastructure.

III. Challenges and Mechanisms for Resource Elasticity Provisioning

In this section, we provide a set of ideas on how to provision resource elasticity, in particular the technical challenges in the virtualized architecture of 5G systems that resource elasticity is meant to address, as well as design hints on the type of solutions or mechanisms that could address those challenges. Table I provides a summary of the content of this section.

A first challenge in virtualized networks is the need to perform graceful scaling of the computational resources required to execute the VNFs according to the load. In that respect, the computational elasticity innovation refers to the ability to scale NFs and their complexity based on the available resources: In case of resource outage, NFs would adjust their operation to reduce their consumption of computational resource while minimizing the impact on network performance.

The second challenge can be illustrated with the current LTE design of the protocol stack, where the NFs co-located in the same node are inter-dependent, *i.e.*, interact and depend on each other. One example of logical dependencies within the stack is the recursive interaction between Modulation Coding Scheme, Segmentation, Scheduling, and RRC. In addition to logical dependencies, traditional protocol stacks also impose stringent temporal dependencies, *e.g.*, the Hybrid Automatic Repeat Request (HARQ) requires a receiver to send feedback informing of the decoding result of a packet within 4 ms after the packet reception. Indeed, traditional protocol stacks have been designed under the assumption that certain functions reside in the same (fixed) location and, while they work close to optimality as long as such NFs are co-located in the same node, they do not account for the possibility of placing these NFs in different nodes. To deal with this challenge, a new protocol stack, adapted to the cloud environment, needs to be designed. This new protocol stack relaxes and potentially removes the logical and temporal dependencies between NFs, with the goal of providing a higher flexibility in their placement. This elimination of interdependencies among VNFs allows the orchestrator to increase its flexibility when deciding where to place each VNF, hence the name orchestration-driven elasticity.

A final challenge of the envisioned 5G architecture appears at the intersection of virtualization and network slicing, *i.e.*, the

TABLE I
INNOVATION AREAS, CHALLENGES AND POTENTIAL SOLUTIONS TOWARDS AND ELASTIC 5G ARCHITECTURE.

Innovation Areas	Challenges	Potential Solutions
Computational elasticity	Graceful scaling of computational resources based on load	Elastic NF design and scaling mechanisms
Orchestration-driven elasticity	NF interdependencies	Elastic cloud-aware protocol stack
Slice-aware Elasticity	E2E cross-slice optimization	Elastic resource provisioning mechanisms exploiting multiplexing across slices

need for end-to-end (E2E) cross-slice optimization such that multiple network slices deployed on a common infrastructure can be jointly orchestrated and controlled in an efficient way while guaranteeing slice isolation. To address this challenge, it is important to devise functions that optimize the network sizing and resource consumption by exploiting statistical multiplexing gains. Indeed, due to load fluctuations that characterize each slice, the same set of physical resources can be used to simultaneously serve multiple slices, which yields large resource utilization efficiency and high gains in network deployment investments, as long as resource orchestration is optimally realized.

We now explain in detail each of the identified innovation areas.

A. Computational Elasticity

The goal of exploiting computational elasticity is to improve the utilization efficiency of computational resources by adapting the NF behavior to the available resources without impacting performance significantly. Furthermore, this dimension of elasticity addresses the notion of computational outage, which implies that NFs may not have sufficient resources to perform their tasks within a given time. In order to overcome computational outages, one potential solution is to design NFs that can gracefully adjust the amount of computational resources consumed while keeping the highest possible level of performance. RAN functions in particular have been typically designed to be robust only against shortages on communication resources; hence, the target should be directed at making RAN functions also robust to computational shortages by adapting their operation to the available computational resources. An example could be a function that chooses to execute a less resource-demanding decoding algorithm in case of resource outages, admitting a certain performance loss.

In addition, the scaling mechanisms, *i.e.*, the modification of the amount of computational resources allocated to such computationally elastic NFs may help in exploiting the elasticity of the system if they are properly designed. There are two significant ways to scale a NF: (i) horizontal scaling, where the system is scaled up or down by adding or removing new identical nodes (or virtual instances) to execute a NF, and (ii) vertical scaling, where the system is scaled out or in by increasing or decreasing the allocated resources to the existing node (or virtual environment) [10]. As an example in the RAN domain, supporting higher system throughput by adding additional access points is referred as horizontal scaling, whereas an increase in operating bandwidth is referred as vertical scaling.

B. Orchestration-driven Elasticity

This innovation focuses on the ability to re-allocate NFs within the heterogeneous cloud resources located both at the central and edge clouds, taking into account service requirements, the current network state, and implementing preventive measures to avoid bottlenecks. The algorithms that implement orchestration-driven elasticity need to cope with the local shortage of computational resources by moving some of the NFs to other cloud servers which are momentarily lightly loaded. This is particularly relevant for the edge cloud, where computational resources are typically more limited than in the central cloud. Similarly, NFs with tight latency requirements should be moved towards the edge by offloading other elastic NFs without such tight timescale constraints to the central cloud servers.

To efficiently implement such functionalities, special attention needs to be paid to (i) the trade-off between central and edge clouds and the impact of choosing one location for a given function, and (ii) the coexistence of Mobile Edge Computing (MEC) and RAN functions in the edge cloud. This may imply scaling the edge cloud based on the available resources, clustering and joining resources from different locations, shifting the operating point of the network depending on the requirements, and/or adding or removing edge nodes [11].

C. Slice-aware Elasticity

Finally, this section addresses the ability to serve multiple slices over the same physical resources while optimizing the allocation of computational resources to each slice based on its requirements and demands, a challenge earlier referred to as E2E cross-slice optimization. Offering slice-aware elastic resource management facilitates the reduction of CAPEX and OPEX by exploiting statistical multiplexing gains. Indeed, due to load fluctuations that characterize each slice, the same set of physical resources can be used to simultaneously serve multiple slices, as Fig. 2 illustrates.

Adaptive mechanisms that exploit multiplexing across different slices must be designed, aiming at satisfying the slice resource demands while reducing the amount of resources

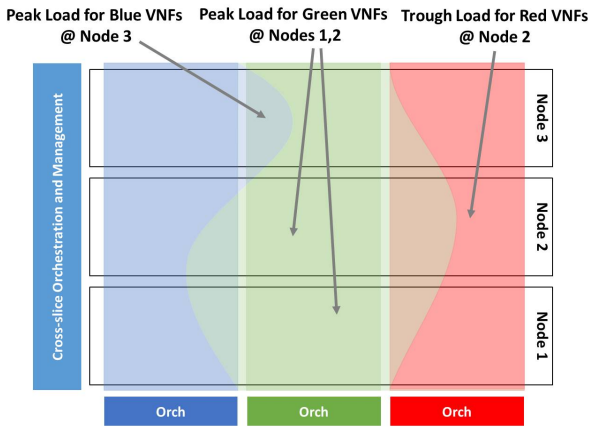

Fig. 2. Illustration of slice-aware elasticity.

required. Hence, the solutions must necessarily dynamically share computational and communications resources among slices whenever needed. An elastic admission control system would be also required, as elastic slices need not have the same amount of available resources as *e.g.*, a highly resilient slice where all resource demands must be fully satisfied at each point in time. Furthermore, in this context a monitoring module should be deployed to retrieve the information required to take optimal sharing decisions, considering trust relationships issues for slices managed by different tenants. To illustrate the above requirements, we provide the high-level sketch of an algorithm for slice-aware elastic resource management consisting of the following key steps:

1) *Forming the available resource pool:* In the first step, the algorithm has to identify the available physical resources and form the shared resource pool for the serving slices. Based on the slices requirements and their SLAs, the algorithm allocates the available computational resources to each slice.
2) *Estimating the total computational capacity:* In this step, the algorithm maps the total computational capacity to the slices requirements (*e.g.*, NF processing time as a function of input variables such as the allocated number of Physical Radio Blocks in the RAN). The admission controller could use the output of this step to decide whether to admit any new slice in addition to determining the service level each slice can receive.
3) *Allocating the available computational resources to different slices:* The algorithm allocates the required computational resource to the NFs of each slice ensuring the total processing time is acceptable. The allocation procedure should consider SLAs type and slices priorities [12].
4) *Observing the network performance and re-allocating the computational resources based on the changes of demands:* The resource management algorithm observes the changes on the network performance as a result of the changes to the resource demands or resource availability, and updates the resource allocation accordingly.

Finally, Artificial Intelligence (AI) and big data analytics are positioned as key enablers to characterize and process this information to make well-informed complex orchestration decisions. While AI-based techniques, in particular machine learning, allows model-free optimal policy derivations for resource allocation mechanisms, smart resource assignment algorithms should know, analyze and react based on the real consumption data provided by big data analytics.

IV. Architectural Components for Resource Elasticity

Many of the ongoing efforts to define a 5G architecture use a four-layer functional structure similar to the one depicted in Fig. 3 initially envisioned by the 5G-MoNArch project [13]. The Service Layer, at the top of this structure, comprises business-level decision functions, applications, and services, operated by a tenant or other external entities. Such functions and services are applied to the network through operations in the Management & Orchestration Layer. This layer provides a multi-tenant, multi-service environment that enables E2E service and resource orchestration. Similarly to the ETSI NFV MANO architecture, the Management & Orchestration Layer in Fig. 3 incorporates components that deal with the life cycle management of the virtual resources (Virtual Infrastructure Manager (VIM)), the life cycle management of the VNFs (VNF manager) and the overall orchestration of the resources and the services on top of those managers (NFV-Orchestrator (NFV-O)). Additionally, it includes slice-aware and domain-specific entities to manage the functional part of the VNFs.

The Management & Orchestration Layer further utilizes a Control Layer, which accommodates, using the intra- and cross-slice controllers based on SDN principles (ISC and XSC in Fig. 3), the required translation of the northbound management and orchestration services into commands that are applied to the actual VNFs and PNFs. The VNFs and PNFs compose the lower layer in the reference architecture, referred to as Network Layer. In order to abide by the fundamental 5G direction for multi-tenancy support on top of a softwarized and slice-enabled network, the Network Layer incorporates separated control-plane and user-plane NFs, which are further divided into slice-specific NFs and shared NFs among different slices.

Within each of these constituent layers of the 5G system architecture shown in Fig. 3, several components are essential to provide and/or exploit elasticity in the system. In particular, the following components should be highlighted:

- *Elastic NFs:* As explained in Section III-A, VNFs can be (re-)designed with elastic principles in mind such that (i) the computational resources available for its execution are taken into account, or (ii) its temporal and/or spatial interdependencies with other VNFs are removed, hence allowing its orchestration to be much more flexible when deciding for a location for its execution. From the above it follows that the existence of this type of NFs is needed to exploit computational elasticity as defined in Section III-A.

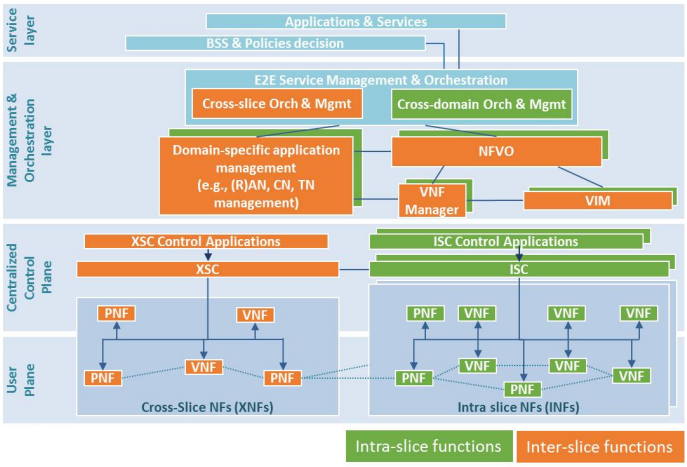

Fig. 3. 5G baseline architecture for 5G-MoNArch as defined in [13].

- *Elastic intra-slice orchestrator:* Elasticity-aware algorithms are needed to orchestrate the different NFs that are part of the same slice. The tasks of such an elasticity-aware orchestrator may include re-locating NFs (from central to edge cloud and vice versa, or from one server to another) depending on available resources, horizontally or vertically re-scaling the amount of resources allocated to one particular NF or a set thereof, clustering and joining resources from different locations, etc. Hence, this module would be responsible for implementing the dimension of elasticity described in Section III-B.
- *Elastic cross-slice orchestrator:* The cross-slice orchestrator is in charge of performing the management and control of the multiple slices that share the architecture, *i.e.*, enabling slice-aware elasticity as described in Section III-C. Some, or all of these slices may be elastic, *i.e.*, slices that do not have totally stringent requirements but rather admit graceful degradation. For those cases, specific orchestration algorithms need to be designed.
- *Elastic controller:* The SDN-like centralized controller is responsible for carrying out the control of the elastic VNFs within a slice, as well the shared elastic VNFs across slices, ensuring a correct multi-tenant operation. This is done through applications that run on top of the controller and implement the logic of the elastic VNFs.

In addition, all the above elasticity-related functionalities could be greatly enhanced with an AI-based engine similar to the one recently being proposed by the Experiential Networked Intelligence (ENI) ISG of ETSI [14]. Focused on optimizing the operators experience, this engine would be equipped with big data analytics and machine learning capabilities that could enable a much more informed elastic management and orchestration of the network, often allowing proactive resource allocation decisions based on the history rather than utilizing reactive approaches due to changes in load. For example, reinforcement learning algorithms could be very suitable to determine optimal policies for horizontal or vertical scaling decisions of NFs, or better slice orchestration decisions could be made if real utilization data is gathered and processed from the underlying infrastructure. The detailed specifications of such a module including the particular algorithms it would apply as well as the description of its interfaces and data collection requirements are beyond the scope of this paper. Nevertheless, an important part of the future work planned within the 5G-MoNArch project focuses on this exact issue.

V. Conclusions

In the quest to dramatically increase the flexibility of networks, in this paper we have introduced the concept of resource elasticity for 5G network architecture. In addition to providing a definition, set of requirements and KPIs, we proposed the exploitation of elasticity along three different dimensions: computational elasticity, orchestration-driven elasticity, and slice-aware elasticity. Challenges and mechanisms for resource elasticity provisioning have been pointed out in each of the dimensions. Finally, we provided a brief overview of the elasticity implications for the main architectural components of a 5G system.

Acknowledgment

Part of this work has been performed within the 5G-MoNArch project, part of the Phase II of the 5th Generation Public Private Partnership (5G-PPP) program partially funded by the European Commission within the Horizon 2020 Framework Program.

References

[1] 5G-PPP, "5G Empowering Vertical Industries," white paper, Feb. 2016.
[2] Next Generation Mobile Networks (NGMN) Alliance, "5G White Paper", Feb. 2015, Available Online: https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf
[3] J. G. Herrera, and J. F. Botero. "Resource allocation in NFV: A comprehensive survey" *IEEE Transactions on Network and Service Management,* vol. 13, no. 3, pp. 518-532, Sept. 2016.
[4] N. R. Herbst, S. Kounev, and R. H. Reussner, "Elasticity in Cloud Computing: What It Is, and What It Is Not", in *Proc. of International Conference on Autonomic Computing (ICAC),* vol. 13, pp. 23-27, Jun. 2013.
[5] E. F. Coutinho *et al.*, "Elasticity in cloud computing: a survey," *Annals of Telecommunications,* vol. 70, no. 7-8, pp. Aug. 2015.
[6] N. Nikaein, M. Marina, S. Manickam, A. Dawson, R. Knopp and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM Computer Communication Review,* pp. 33-38, Oct. 2014
[7] P. Rost *et al.*, "Mobile network architecture evolution toward 5G," *IEEE Communications Magazine,* vol. 54, no. 5, May 2016.
[8] 5GPPP Vision Brochure, Available Online: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf
[9] EU H2020 project 5G-MoNArch, Deliverable D6.1, "Documentation of Requirements and KPIs and Definition of Suitable Evaluation Criteria," Sept. 2017.
[10] B. Walder, "Cloud Architecture Patterns," O'Reilly Publications, 2012.
[11] J. Oueis emphet al., "Distributed mobile Cloud Computing: A multi-user Clustering Solution" in *Proc. of IEEE Int. Conf. on Communications (ICC 2016),* 23-27 May, 2016, Kuala Lumpur, Malaysia.
[12] S. Khatibi and L. M. Correia, "A model for virtual radio resource management in virtual RANs," *EURASIP J. Wirel. Commun. Netw.,* no. 1, pp. 68, Mar. 2015.
[13] EU H2020 project 5G-MoNArch, Deliverable D2.1, "Baseline architecture based on 5G-PPP Phase 1 results and gap analysis," Oct. 2017.
[14] ETSI ENI - Experiential Network Intelligence, Available Online: https://portal.etsi.org/Portals/0/TBpages/ENI/Docs/ETSI_ISG_ENI_Presentation.pdf