Computer Communications 179 (2021) 131–144

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

NASOR: A network slicing approach for multiple Autonomous Systems

Rodrigo Moreira [a,b,*], Pedro Frosi Rosa [a], Rui Luis Andrade Aguiar [c], Flávio de Oliveira Silva [a]

[a] Faculty of Computing (FACOM), Federal University of Uberlândia, Uberlândia, 38400-902, Brazil
[b] Institute of Exact and Technological Sciences, Federal University of Viçosa, Rio Paranaíba, 38810-000, Brazil
[c] Telecommunications Institute (IT), University of Aveiro, Aveiro, Portugal

A R T I C L E I N F O

Keywords:
SDN
NFV
Segment rouging
Network slicing
Multi-domain
Orchestration
Recursive Network Slicing

A B S T R A C T

Realizing network slicing inside and between Autonomous Systems (ASs), that is, multi-domain, is challenging because there is no consensus or solutions that consider both policy and technological independence between domains. Many approaches found in the literature aimed to realize network slices that span across multiple ASs. However, they commonly rely on cross-connected technologies or domain-coupled such as Virtual Private Network (VPN) or Multi-Protocol Label Switching (MPLS). This work addresses the issue of multi-domain network slicing by leveraging technologies such as Software-defined Networking (SDN), Segment Routing (SR), and Network Functions Virtualization (NFV) in an innovative distributed framework, called Network And Slice ORchestrator (NASOR). Our work advances resource management and orchestration potentialities, providing a recursive network slice mechanism and adding dynamism in the network slice deployment between multiple domains through an open interface. As a result, NASOR functionally outperforms its peers. Experiments showcased the proposal's applicability and scalability in multi-domain network slicing. Additionally, experiments suggest that an open interface enhances network slices' customization degree and improves the network Quality of Service (QoS) in typical Internet applications, such as Voice over Internet Protocol (VoIP).

1. Introduction

The number of Internet users worldwide will continue its growth, and by 2023 a total of $5.3bn$ subscribers will have fixed access with a speed of at least 110 Mbps [1]. This traffic will aggregate in the transport/core of the network, which needs to support application requirements. Considering this huge amount of connections, the network needs to have automated management tools to provide custom connectivity to applications. Network slicing has received significant attention recently due to its ability to offer tailored resources to applications.

A network slice is a portion of shared compute and network resources that support connectivity between entities. The network slice is manageable and has isolated management, control, and data planes. In the context of this work, this connectivity occurs across the Internet between multiple Autonomous Systems (ASs) [2,3]. Network slicing, leveraged by network programmability, is decisive to offer tailored network resources to a wide range of applications [4]. To the users, network slicing offers flexibility, and dynamic management of their connectivity [5].

Problem statement. Previous network slicing solutions tried to realize resource sharing standing on cross-connected technologies such as Virtual Private Network (VPN) or domain-coupled like Multi-Protocol Label Switching (MPLS) or Software-defined Networking (SDN). Similarly, approaches using Virtual LAN (VLAN), Generic Routing Encapsulation (GRE), Virtual Extensible LAN (VxLAN), Network Service Header (NSH) claimed network slicing realizations [6,7]. However, to realize network slicing requirements, all these solutions lack some form of network programmability, isolation, management of control and data planes, and a distributed mechanism [7,8] able to cope with multiple domains.

To the best knowledge of the authors, there are no well-known solutions capable of establishing network slice connectivity inter-ASs, except in particular domains such as data centers [9] or mobile network [10]. The multi-domain or inter-ASs network slicing is not currently fully addressed because the existing solutions avoid dealing with the Internet data plane. The Internet data plane is the datapath built by the routing algorithms, such as Border Gateway Protocol (BGP), enabling end-to-end connectivity between endpoints.

The observed trend is that the new mobile network architectures drove current slicing network proposals and technologies [3]. Consequently, those approaches focused exclusively on the data and control planes of the core or accesses of mobile networks [11,12]. However, the compute and network slicing must be systematically exploited at other system hierarchy levels [13] to be able to cater to both user

* Corresponding author at: Institute of Exact and Technological Sciences, Federal University of Viçosa, Rio Paranaíba, 38810-000, Brazil.
E-mail addresses: rodrigo.moreira@ufu.br, rodrigo@ufv.br (R. Moreira).

https://doi.org/10.1016/j.comcom.2021.07.028
Received 19 August 2020; Received in revised form 21 June 2021; Accepted 29 July 2021
Available online 5 August 2021
0140-3664/

and application requirements, all under the control of different domain operators.

Proposed solution. Thus, addressing network slicing between multiple domains requires a distributed mechanism to work across these different administrative domains. Therefore, following network slicing realization boosted in the 5G specification, our work goes beyond network slicing to transport networks. Hence, our proposal deals exactly with the network slicing between multiple domains. Also, it embodies the recursive feature to the multi-domain, an area still without standardization and in its early implementation stages [14,15].

This paper presents the ***N**etwork **A**nd **S**lice **OR**chestrator* (NASOR), a hierarchical and distributed framework capable of deploying network slicing inside and between ASs. It can also provide recursive network slicing for Virtualized Everything Functions (VxFs) connectivity spanning multiple domains. Leveraged by SDN, NFV, and Segment Routing (SR) technologies, *NASOR* functionally outperforms related works, which aimed to realize multi-domain network slicing. *NASOR* materialized a management and orchestration framework for network slice deployment between multiple ASs.

Our multi-domain network slice offers end-to-end connectivity between endpoints hosted in different ASs. This connectivity adaptively supports both application and user requirements. The network slicing mechanism grants the user direct management capabilities using separate data and control planes. Thereby, *NASOR* filled the state-of-the-art gap concerning the programmability, deployment, and management of multi-domain network slices, offering a standardized management interface, the Open Policy Interface (OPI).

In a nutshell, the main contributions of this paper are listed as follows:

- A framework to slice networks and compute resources recursively;
- A mechanism for establishing logical connectivity, designed as network slices, across the multiple ASs;
- An open interface designed to handle fine-grained requirements description and dynamism in establishing network slices;
- A mechanism for advertising network and computational capabilities for multiple ASs;
- A performance assessment of *NASOR* considering scalability and performance;
- A short survey of state-of-the-art approaches aimed at providing resource slicing.

The remainder of this paper is organized as follows. Section 2 presents a short survey of related works and points out the contribution of our approach. Section 3 presents the *NASOR* framework designed to meet state-of-the-art challenges, and at the end, we proposed mapping of NASOR to the main standards. Section 4 presents a scenario for slicing resources and shows its applicability and adequacy to face multi-domain challenges. Section 5 presents the results and statistical assertions of the experiments. Finally, Section 6 presents the final remarks, lessons learned, and research directions, followed by references.

2. Related work

This section presents approaches that relate to and underlie our solution. In previous work [16], we introduced essential *NASOR* components and evaluated them. However, this paper presents a complete innovation with a new feature explanation and new use-case experiments. It is structured evolutionarily, punctuating the approaches that provide resource sharing as service slices. Some are closer to computational resource slicing in the proposals described below, others to network slicing, and others consider both approaches simultaneously. In our case, resources refer to the physical network, storage, and compute facilities placing on domains.

The likely separation in previous approaches was overcome in current approaches leveraged by SDN and Network Functions Virtualization (NFV) technologies to make network and computing management simultaneous and holistic [17]. At the end of the section, we summarized some approaches contrasting them against our proposal in a table. In the next sections, we describe and group some related work as Open-source Orchestrators, Research Projects, and Research Contributions.

2.1. Open-source orchestrators

Some initiatives, mostly open-source, towards Network Service orchestration, became essential to enable innovation in the network slicing ecosystem. Following, we describe a short survey containing some of these.

The Extensible Service ChAin Prototyping Environment (ESCAPE) project materializes in an orchestrator the vision of the UNIFY [18] project from 2013. This project relates to network services orchestration in the multi-domain scenario. The architecture has three interconnected layers, namely: Service Layer, Orchestrator Layer, and Infrastructure Layer. The Service Layer is the interface that receives the service request and forwards it to the Orchestration Layer, which evaluates the request and allocates resources for the service. The Infrastructure Layer contains the mechanisms for managing network and computational resources. These interconnected layers provide the functionality to ESCAPE's architecture, besides ensuring the management of the services that run on it [19].

The Open Source MANO (OSM) project from 2016 [20] is an open-source orchestrator from European Telecommunications Standards Institute (ETSI). OSM provides Virtualized Network Functions (VNFs) life-cycle management. The OSM architecture has three components: Network Function Virtualization Infrastructure (NFVI), MANO, and Network Service. NFVI hosts virtual machines and connects them through virtual links. MANO maintains the configuration and manages the life cycle of the VNFs and Network Slices (NS). The third block is the collection of VNFs, Network Service, and Network Slices combined and interconnected in NFVI to materialize the service instance. OSM has a service architecture based on containers that bring modularity. Also, OSM provides connectivity for VNFs in a data plane based on SDN [20]. From a service management perspective, OSM is more focused on computational resources. However, mechanisms are available for slicing network and computing resources across data center domains in this ecosystem.

On the other hand, the Open Network Automation Platform (ONAP) proposal offered a platform for managing the life-cycle of new services to developers and service providers [21]. The ONAP guides are real-time orchestration, policy-oriented, and automation of physical and virtual resources. The platform is the union of two approaches, Management and Orchestration (MANO), OPEN-O, and Open-ECOMP [22]. This union decouples the details of technology services through standardization models information, general management, and central orchestration platform. ONAP provides the deployment and management of network services, using big data and artificial intelligence as enabling technologies. The deployment of services through ONAP can take place on the Internet Service Provider (ISP) or in their private cloud.

2.2. Research projects

Many standard bodies, industries, academies, and associations carried out initiatives to provide models, and research findings to compose 5G building-block, especially for network slicing realization [2,23,24]. We describe some of these research projects, highlighting their architectural framework and implementation.

The SONATA architecture, from 2015, complied with the framework ETSI and met the demand for a consistent and integrated solution for the complete life-cycle management for virtualized network services. SONATA is a proposal for a service architecture and orchestration structure to develop virtualized services conceived as resource slices. SONATA carries two components, the Software Development Kit (SDK) and Service Platform (SP). The model and programming tools implement the services using the DevOps approach. There is a public catalog, where it stores artifacts, such as manifest files that describe functions and services. On the other hand, SP integrates SONATA with interfaces for platform operators and service developers and allows the deployment of services in different infrastructures. The interaction between SDK components and infrastructure is carried out through the Virtualized Infrastructure Managers (VIMs) management, which runs the services [25].

In 2015 the 5G-Crosshaul project raised a transport architecture for mobile networks. This architecture bases on flexibility and efficiency requirements for resource allocation in the NFV specification. Its objectives include a network slice implementation according to the "as-a-service" model through the programmability principles supported by SDN. Concerning the control mechanism, instantiation, and resource placement, the authors explore the concept of virtual service infrastructure, which allowed the use case, such as virtual mobile networks. The authors support the novelty on top of its ability to provide service recursion for multi-tenants since one service instance can instantiate another [26].

In 2016, the 5GEx project, inspired by IPExchange [27], proposed to contribute with a service orchestration mechanism from a multi-domain perspective. Its architectural design comprises a high-level Orchestrator, which deals with the multi-domain aspect and provides an interface for service management. The 5GEx Orchestrator deals with managing networks, slicing, and virtual machine management in different domains by exchanging messages between the Orchestration components. The solution architecture foresees that Orchestrators are located in each domain to allow the slicing of resources in logical networks and the centralized management of each domain over its resources [28].

In 2016, the 5G!Pagoda, an EU-Japan project, brought advances in resource slicing within mobile networks, specifically, smart city verticals. The solution is realized with an architecture and a framework to provide network slices with programmable characteristics and scalable management. This project built testbeds to cope with the challenges described above, especially for experimentation with climate applications and safe societies. The solution architecture contains a hierarchical orchestration model. Each technological domain has its resource manager, which is associated with a general orchestrator, who performs multi-domain management. Hence, the solution architecture has three interconnected blocks: Multidomain Slice Orchestrator, Commercial Service Slice Orchestrator, Domain-Specific Slice Orchestrator [29].

Similarly, the 5GTransformer project, from 2017, built a testbed for experimentation based on verticals for mobile networks for automotive, entertainment, eHealth, Industry 4.0, and Virtual Mobile Networks. The project has considered the multi-domain orchestration model based on federation, where each infrastructure, geographically distributed, connects to provide the requested services. The main testbed components are the vertical slicer, vertical orchestrator, transport management platform, and computing resource for mobile networks. In addition to these three main components, the solution architecture contains a monitoring mechanism that runs within the three and collects metrics to guarantee service agreements and quality of service. The architecture also provides interfaces for operation support and for receiving the experimentation verticals. In the architectural model, the inter-domain data plane provides encapsulation through technologies, such as VLAN, VxLAN, MPLS, VPN, and others, for data exchange between administrative and technological domains–data center [30].

The Katana framework developed within the scope of the 5GENESIS Project [31], from 2018, aimed to realize network slicing at edge [32]. The Katana components comprise entities and roles such as North Bound Interface API, Slice Mapping, Slice Provisioning, Slice Monitoring, and Adaptation Layer. The North Bound Interface API provides life-cycle management to the experimenter or Slice Manager. The Slice Mapping entity provides optimal selection mechanisms for allocating resources to new slices. Slice Provisioning is responsible for maintaining Wide Area Network (WAN) paths and radio components. The Slice Monitoring entity's role is to handle the monitoring and welcome the slice of the network deployed. Finally, Adaption Layer provides an abstraction for the technological domain that translates messages into formats supported by the components. Carried experiments considering service deployment Key Performance Indicators (KPIs) contrasted Katana against OSM-based network slicing deployment on edge.

The Novel Enablers for Cloud Slicing (NECOS) project from 2018 stands on the Lightweight Slice Defined Cloud (LSDC) concept. It materializes a slice-as-a-service approach that spans multiple cloud computing infrastructures. NECOS aims to address the challenges of deploying applications and services by network operators and service providers. Deployment challenges include energy efficiency, versatility, security, and availability of resources. Besides, the challenges of traditional cloud computing have leveraged the NECOS deployment model to address stability challenges. Its design features included cloud and network management, service orchestration, and monitoring of low-cost distributed resources [9].

In 2018, the 5GinFIRE proposal project brought an experimentation ecosystem based on the ETSI NFV reference model for deploying applications in the 5G mainstream. The project addressed the challenges of a holistic and unified view of a testbed for vertical experimentation and as an ecosystem of tests for hosting and integrating applications to meet each vertical's suitable requirements. The solution architecture consists of computational resources geographically distributed and interconnected in a hub-and-spoke format for the central control entity. The project established the vertical portal concept for users to build, deploy and experiment with VNFs and NSs. The 5GinFIRE has advanced state-of-the-art as a testbed, incorporating the ETSI MANO structure's scalability to define and describe an experiment, taking into account the applications as industry verticals [33].

In 2018, the Platform for Open Wireless Data-driven Experimental Research (POWDER) [34] was proposed in the US. POWDER addressed resource sharing and evolved a novel applications experimentation ecosystem. POWDER is a city-scale platform for advanced wireless experimentation such as novel advanced wireless broadband and communications technologies. The highly programmable and flexible features have driven the platform to offer multi-input multi-output, a key enabler for 5G networks and beyond. POWDER has three main components: physical infrastructure where facilities are built, the functionality that combines hardware and software to realize the infrastructure's functionality, and the control framework that manages resources and provides experimentation services to users. The hardware layer of POWDER has radios equipment, base stations, and computing interconnected by dedicated fiber in downtown areas, residential, and a hilly campus environment. Also, the software layer includes well-known general-purpose network virtualization and cloud computing stacks and various Software-defined Radio (SDR) stacks.

2.3. Research contributions

There are many efforts and Proofs-of-Concept (PoC) which aimed to offer multi-domain network slicing. We considered in this category papers ranging from mobile networks to more generic ones. Besides, there are papers out of the context of 5G Projects or consortiums. In this section, we will present some of these approaches, highlighting their leading technologies and concepts.

The 5G Cross-Domain framework, presented in [15], aimed to provide network slicing between separate administrative domains, combining four components. First, the Service Broker handles slice creation requests across multiple domains and manages a global repository to support service deployment. The Service Conductor component provides management between federated domains. The third component, whose behavior is similar to an orchestrator, allocates internal resources from each domain in the slicing establishment and deals with life-cycle management. The third component has four minor functionalities: service management, slice life-cycle management, management and orchestration for subdomains, and SDN controller for subdomains. The fourth component outlines the subdomains' infrastructure: VNFs, virtual resources, virtualization layer, and physical infrastructure.

A 5G Framework is presented in [35]; this framework aimed to design and operate network slices with flexibility, automation, and collaboration. The proposal subdivides the solutions' implementation into three blocks: Definition, Solution, Scope, Instantiation, and Service Update. The first one concerns the phase in which the service characteristics refer to "what" are defined through a document containing the service description. The second, Service Resolution, is the phase to define connections, hardware and software adjustments, and the visualization of the service chain. Also, it contains the "how-to" for instructing the interfaces, characteristics, resources, and assessment of the service description. Finally, the Scope of Service comprises the translation of the specifications from design to deliver the service scope.

The PERMIT work materializes resource slicing on mobile networks to provide different levels of granularity, including by network, application, user group, individual users, and user data. Similar to their peers, their approach combines network programmability concepts with cloud computing as enabling technologies for the framework. The solution has two orchestrators: the Mobile Network Personalization Service Orchestrator (MNP-SO) and Mobile Service Personalization Service Orchestrator (MSP-SO), which has mechanisms for personalizing the service and mobile network for users. The architecture leveraged by the framework NFV advances the state-of-the-art with an orchestrator of functionalities aimed at creating virtual mobile networks [36].

A recursive resource slicing approach described in [37] proposes a recursive method for computing and network slicing. The proposal, leveraged by framework NFV, allows users to create and manage slices exclusively by a private MANO. They also offer MANO-as-a-service (MANOaaS) architecture and the distributed negotiation process for managing the levels of agreement between the infrastructure and the tenants. Through simulations in the MATLAB tool, the authors demonstrated the success rate in response to the implementation of resource slices, deploying slices over the infrastructure. Empirical results suggest that a higher percentage of response requests are serving when using their approach in the face of the partial approach and completely unmanaged.

A Multidomain Optimizer approach, proposed in [38], provides an optimization-oriented mechanism for managing network functions across multiple domains. In this work, the authors formulated and experimented with an optimization algorithm that should provide the best chaining, considering the maximization of energy efficiency and given a set of mathematical restrictions. The optimization problem defines that, as the requests for the implementation of computational services arrive, under a defined probability, the solution must provide the best mapping of the requests to the computational resources throughout a multi-domain scenario. Additionally, in the proposed optimization model, connectivity was observed between virtualized entities and their best chaining across several domains.

2.4. Discussion

In Table 1 we summarized and organized some state-of-the-art approaches as Open-source Orchestrators, Research Projects, and Research Contributions. These approaches aimed to materialize in some way network slice deployment over shared compute and network resources.

Multi-domain Data Plane. It defines whether the path builder engine handles the parameters configuration across multi-domain entities. This engine must provide logical configurations and service agreements honoring service metrics, even spanning other ASs. Some approaches provide a data plane multi-domain capable [28,30], but most of them do not [25,26].

Multi-domain Control Plane. It relates to handling control routines in network slice deployment and configuration spanning multiple ASs. This column concerns the ability to handle network slice orchestration and management distributedly across multi-domains. Some solutions offer this feature in specific domains such as data center or mobile network [15,36], others in Transport Network (TN) [28,38].

Control Architecture. It regards the solution building-block architecture. Most solutions are monolithic [25], implying a single point of failure. Others hierarchical [9], raising doubts about each domain's logical and functional independence. Lastly, distributed network slicing architectures are known [28] which, although essentially distributed, do not offer a distributed data plane.

Network Placing. It characterizes the solutions regarding their effectiveness and ability to influence end-to-end entities on Transport Network (TN). Some solutions handle network slicing in the data center [21,29] implying in limited technological influence domain. Others provide network slicing at the network core or in mobile networks' access, and their entities [15,36].

Slicing on top of the Internet. It classifies the slicer engine regarding its ability to pave a data plane on top of the Internet entities. Many network slicing proposals use technologies for datapath pavement such as VPN [29,33], encapsulation [39], or restricted to a single domain such as SDN-based approaches [37].

Inter-Domain Data Exchange. It concerns if solutions ensure the multi-domain network slicing across multi ASs. Some solutions address this separately, enabling network slicing in the network access [15], core [38], others in back-haul [40], setting aside cross-domain network slicing.

Enabling Technologies. It aims to describe the technological framework or the prevailing technology that the approach had included in its architecture. Most solutions use SDN capabilities [19,29]. Others use NFV [37] or specif frameworks to handle network slicing [35].

Legacy Network Compatibility. It characterizes the solutions regarding their embodiment in an ISP or AS without requiring significant changes in their network implementation and operation format. For some solutions, from a deployment perspective, the slicing approach requires considerable modifications at the many technological levels of the domains, such as the core, WAN, and access [41].

ETSI MANO Compatibility. It identifies whether the network slicing solution is compatible with the well-established ETSI MANO. Some solutions provide their management and orchestration mechanisms compatibility coupled in its domain [9,21]. Our multi-domain network slicing vision concerns openness and inter-working with standardized and widely accepted solutions.

Service Chaining. It describes whether network slicing solutions provide a compute or network service within an established multi-domain network slice.

Third-party Interface. It characterizes the network slice engine whether they can create, change or delete specific slice properties without requiring management or orchestration from ISPs or AS manager. Many networks slicing solutions still do not give the user control over their network slices [26,33,42].

Recursive Slicing Establishment. It describes whether solutions can reshape an established network slice resizing it or creating new ones until reaching a physical and logical limit of the primary slice. In resizing is expected to deliver sub-slices to users with separated management, control, and data plane. Although specification frameworks define recursive slicing [43], many solutions do not perform multi-domain network slicing, especially in a recursive way [25,30,36].

Dynamic Service Deployment. It concerns the network slice user/owner enablement to define a network slice datapath and its attributes onward the target AS. It gives the user the ability to set network slicing parameters on intermediate ISP os AS entities, paving an end-to-end and multi-domain datapath. Most solutions do not give the user the freedom to specify their network slice's path and parameters.

Table 1
A short survey of related works.

Approach	SOTA Category	Multi-domain Data Plane	Multi-domain Control Plane	Control Architecture	Network Placing	Slicing on top of the Internet	Inter-Domain Data Exchange	Enabling Technologies	Legacy Network Compatibility	ETSI MANO Compatibility	Service Chaining	Third-party Interface	Recursive Slice Establishment	Dynamic Service Deployment
ESCAPE [42]	Open-source Orchestrator	○	●	Monolithic	N/A	○	SDN Domain	SDN and Cloud	○	○	●	○	○	○
OSM [20]	Open-source Orchestrator	○	●	Monolithic	DC	○	SDN Domain and VPN	SDN and Cloud	○	●	●	●	○	○
5GEx [28]	Research Project	◑	●	Distributed	Core	○	IP Exchange	SDN and Cloud	●	●	●	●	○	○
5G Framework [35]	Research Contribution	○	○	Monolithic	DC	○	SDN Domain	SDN, Cloud, and BPMN	○	●	●	●	○	○
5G Cross Domain [26]	Research Contribution	○	●	Monolithic	DC, and Mobile Network Access	○	SDN Domain	SDN, and Cloud	○	●	●	○	○	○
SONATA [25]	Research Project	○	●	Monolithic	DC	○	SDN Domain	SDN, and Cloud	○	●	●	●	○	○
5GinFIRE [33]	Research Project	○	○	Monolithic	DC	○	SDN Domain, and VPN	SDN, and Cloud	○	●	○	●	○	○
5G!Pagoda [29]	Research Project	○	●	Monolithic	DC	○	Internet, and VPN	SDN, and Cloud	○	●	○	○	○	○
5GTransformer [30]	Research Project	●	●	Distributed	DC	○	VPN Encapsulation	SDN, and Cloud	○	●	○	●	○	○
ONAP [21]	Open-source Orchestrator	○	●	Monolithic	DC	○	Internet, VPN, and SDN	SDN, and Cloud	○	○	●	●	○	○
PERMIT [36]	Research Contribution	○	○	Monolithic	Mobile Network Access	○	Internet	SDN, Cloud, and RAN	○	●	●	●	○	○
NECOS [9]	Research Project	○	●	Hierarchical	Core, and Edge	○	VPN, and SDN	SDN, Cloud, and Containers	○	○	●	◑	○	○
POWDER [34]	Research Project	○	○	Monolithic	DC, Mobile Network Access, Edge	○	L2 and L3 Domain	SDN, NFV, Cloud, SDR, MEC	○	○	○	●	○	○
5G Crosshaul [26]	Research Project	○	○	Hierarchical	Core	○	PBB-TE	SDN, and NFV	○	○	●	○	◑	○
MANO-as-a-Service [37]	Research Contribution	○	○	Monolithic	Core	○	L2 Domain	SDN, and Cloud	○	●	○	●	◑	○
Multidomain Optimizer [38]	Research Contribution	○	○	Hierarchical	Core	○	N/A (Simulation)	SDN, and Cloud	○	○	●	◑	○	◑
Katana [32]	Research Project	○	●	Monolithic	Mobile Network Access and Edge	○	Internet VPN or L2 Domain	SDN, SDR, Cloud, MEC	○	●	●	◑	○	○
NASOR	Open-source Orchestrator	●	●	Hierarchically Distributed	Core	●	Internet	SDN, NFV, Cloud, and SR	●	●	●	●	●	●

3. Network and slice orchestrator

To fulfill the recognized state-of-the-art gap, we propose the ***N**etwork **A**nd Slice **OR**chestrator (NASOR)*, a framework for establishing and managing network slices spanning across multiple ASs. Our framework advances its pairs by relying on the Internet data plane, materializing a recursive slicing mechanism. *NASOR* enables fine-grained requirements specification in the service description file. Also, *NASOR* supports dynamism in network slice deployment, according to user requirements and network utilization state.

The connectivity ecosystem supported by the new network architectures has changed computing and network services to an "as-a-service" mainstream. To answer this challenge, we based the *NASOR* building on the Network Function Virtualization (NFV) framework, precisely the ETSI Architectural Framework [44], including the Network Service view. Considering this ETSI framework, we built a proof-of-concept of the *NASOR* combined with Open Source MANO (OSM) [20] to assess the network slice deployment feasibility across multiple ASs.

3.1. The recursive network slicing

The *NASOR* defines and implements a recursive network slice. Using this feature, the *NASOR* can split a network slice into several child network slices bounded to this parent network slice. Each child slice is manageable independently and it can have other child slices in a recursive way.

Each slice has a virtual forwarding table that enables the multi-ASs datapath to provide connectivity to end-to-end entities. Each slice also has its resource orchestrator, giving the users a management interface to handle the network slice life-cycle. Using this orchestrator, a user can create a new sub-slice that is bounded in the parent slice. This new slice has its virtual forwarding table, resource orchestrator, management, control, and data planes. Using the sub-slice resource orchestrator, a user can recursively create sub-slices. This process stops when the sum of the sub-slices' resources is equal to the parent slices' total resources. After creating a new slice, its resource orchestrator will update its virtual forwarding table until the slice has a route that can reach the final AS. ETSI NFV Architectural Framework does not cover this recursive behavior.

Then, it is essential to map the proposed solution's structure according to the NFV ETSI management and orchestration framework. Thus, the control entities of the present proposal are associated with the entities of the NFV framework. The first mapping aspect in Fig. 1 is to equal the proposal with a structural triad of the framework that enables the management of virtualized network functions.

In this sense, *NASOR* establishes a direct relationship with the components, entities, and roles of the framework, namely: Management and Orchestration (MANO), NFV Infrastructure (NFVI), and VNF Manager (VNFM). Initially, the primary role of *NASOR* is directly associated with that described for MANO, since its purposes are: to manage the service instance layer, that is, the data plane and control of a slice established between domains; manage the physical substrate of domains and resource slices.

For the control plane, the proposed architecture comprises the entity Network and Orchestration (NANO). *NANO* is a user-orchestrator and is a lightweight network slice management orchestrator, instantiated by the domain administrator. *NANO* aims to act as the agent in network slice deployment running and recursively creating new NANOs and forwarding tables. Each *NANO* set up routines and logical path configurations between the Internet route entities. Also, *NASOR* enhances and extends the framework NFV by proposing recursive slicing behavior.

Second, *NASOR* has a well-defined interface with the physical substrate of the domains, which is the interface between management routines and virtualization technology. NFVI and its components are the domain virtualization interface in the proposed mapping. They include: Internet routers, compute resources, virtualization technologies, storage, and network functions.

Therefore, equating the NFVI layer in terms of the roles that its entities play with the structural framework is possible. Thus, the NFVI interface of *NASOR* allows network and computing resource management. *NASOR* plays NFVI management to support the network slice deployment.

Third, it is possible to associate the role of *NASOR* with the entity Virtualized Infrastructure Manager (VIM), foreseen in the architectural framework. This entity guides and manages the interaction between the compute virtualized services and the network substrate. Thus, resource management, allocation, visibility, network services inventory: service slices, and capacity management are handled by the VIM entity

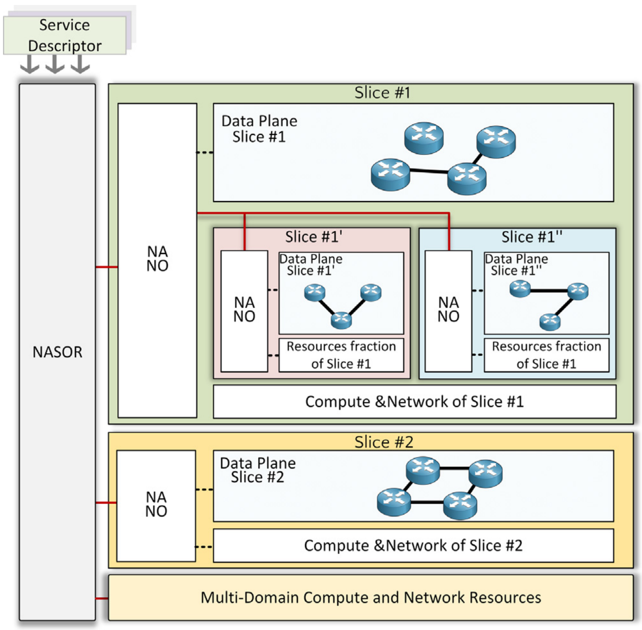

Fig. 1. Proposed solution mapping with the ETSI NFV framework.

of *NASOR*. This proposal advances the state-of-the-art, especially the framework NFV, in the service implementation approach.

The fourth aspect of matching falls on the fact that *NASOR* has an interface for creating and managing slices, similar to the northbound interface. This interface is designed in a graphical user interface format, which guarantees users the service life-cycle management. Thus, *NASOR* proposes that each *NANO* creates and manages the micro-orchestrators life-cycle, including the control plane for recursive network slicing.

The micro-orchestrator's architecture comprises an information repository of the network slices: owner, creation time, specification of the logical separation of networks, network, computational capabilities, and a graphical user interface. Each administrative and technological domain has a *NASOR*, which in turn instantiates, manages, and maintains *NANOs*, which create and manage micro-orchestrators of slices or network sub-slices.

3.2. NASOR: Interfaces and components

The *NASOR* architecture has interfaces and relationships with entities to provide inter-AS network slicing. In the *NASOR* framework, end-to-end refers to the path between multiple ASs, which supports network slices. As shown in Fig. 2, *NASOR* management interface allows the domain administrator to instantiate a network slice, according to their demand, practically and intuitively, through a graphical user interface. This interface ensures the establishment of a network slice to be handled separately in each administrative domain. Each domain has a sub-module of *NASOR*, the Inter-Orchestrator Exchanges (IOEx), which delivers to other ASs the network slicing request. Thus, for a domain to compose a list of those compatible with the slicing, it must have the *NASOR* entity and its sub-components and configurations.

Broker-based Differentiated Services (DiffServ) approaches, which include an entity for inter-domain management, had shown advances in Quality of Service (QoS) enforcement for multi-domain applications [45–48]. This approach had been inspiring distributed-hierarchical architecture of *NASOR*, which in turn shares in addition to the network computational capabilities between domains.

As it is a distributed solution, *NASOR* makes use of persistent data repositories. Initially, the *Orchestrator Information Base (OIB)* stores

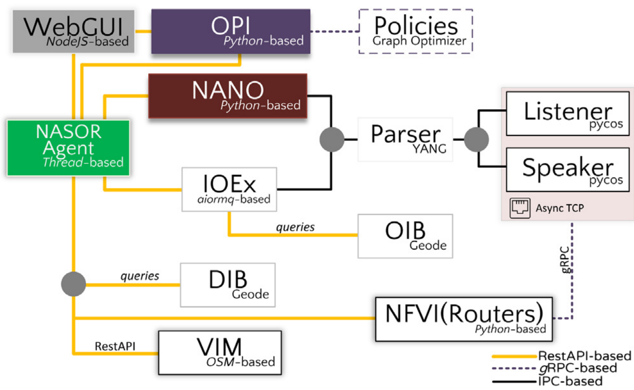

Fig. 2. Interfaces Detailing of *NASOR*.

information of *NASORs* from other domains, including Internet Protocol (IP), Autonomous System Number (ASN), network, and compute capabilities inventory. At this point, each *NASOR* is responsible for updating the reachability information and network and computing capabilities of its domain in the asynchronous repository.

Thus, when *NASOR* is requested to create a multi-domain network slice, it is carried out in the domestic domain, forwarding the request to another domain. A data exchange point between *NASORs* allows exchanging the slice by requesting through the Listener and Speaker routines standardized according to a Parser module.

We have already discussed about the OIB, whose interaction with *NASOR* occurs in a multi-domain slice establishment. However, the Domain Information Base (DIB) allows *NASOR* to store private domain information within the domestic domain. The information includes the routers in the domain by which the network slice stays on top. Also, the DIB repository contains the addressing that ensures VIMs and computational capability reachability.

The *NASOR* agent has an interface with the VIM API that provides the instantiation of computing services across multiple technological domains. This paper chooses OSM due to its versions' current maturity and compatibility with the ETSI framework's management and orchestration. Thus, OSM meets the Local MANO roles, handling compute service deployment in specific domains. However, NASOR deploys network slices crossing multi-domains interacting with local OSMs and playing Multi-domain MANO roles.

3.3. NANO: Specification

The *NANO* architecture has two interface abstractions. One of them is based on the network slice configuration plane, which is directly associated with the *NASOR* routines. This interface handles the first level of the interdomain network slice. The second interface comprises the network slice management plane, designed for the domain administrator. Both interfaces converge to a central entity so that each *NANO* handles containers to implement the micro-orchestrator management.

The left side of Fig. 3 presents the details of the *NANO* operation through its two interfaces. The first interface is based on RestAPI, where *NASOR* requests the remote call of routines for establishing network slices. The second interface provides a graphic management plane for the domain, which allows managing the set of micro-orchestrators related to a network slice, namely: modifying parameters and consulting the state of resources.

According to Fig. 3, our approach provides an interface for the user/third parties to manage their slices. As depicted on the right side of Fig. 3, an Micro-Orchestrator (MO) offers a Graphical User Interface (GUI), which allows the network slice owner to consult and manage parameters of the contracted service. The modification at this level is subject to the resources previously reserved in the initial network

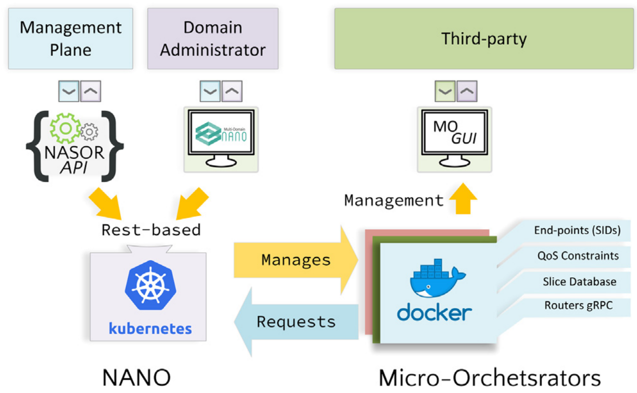

Fig. 3. NANO Interfaces and Components.

slicing process, provided by *NANO*. Hence, this interface offers the network slice owners the possibility to manage the agreed service.

Micro-orchestrator itself is container-based and handles two management levels through Micro-Orchestrator Manager (MOM). The functionality of MOM is to manage micro-orchestrators that run in container form. The inter-domain network slice management is carried out through micro-orchestrator routines. MO orchestrates a network slice by assigning it to a user and persisting the information related to that network slice: end-point SIDs, QoS specifications, network slice database, and router list. Each MO provides a network slice management interface for the user/owner to modify network slice parameters and understand statistics for monitoring purposes.

Fig. 4 depicts the logical sequence in establishing a network slice in multiple domain scenarios. The logical sequence comprises entities such as Network and Slice Orchestrator (NASOR), Orchestrator Information Base (OIB), Network and Orchestrator (NANO), Virtualized Infrastructure Manager (VIM), Network Functions Virtualization Infrastructure (NFVI), and Micro-Orchestrator Manager (MOM).

At this point, it is worth describing two multi-domain approaches: the first refers to a network slice that originates in domain A, and the endpoint of the network slice is in domain B, where it is at a maximum one hope. Thus, intercommunication between domains is direct via traffic exchange.

The second refers to a network slice establishment in which domain B is not a neighbor of the origin domain A. This paper considers that the intermediate domains – conceived as transit domains –, will provide the configurations in their elements, although they are themselves in transit.

The Open Policy Interface (OPI), depicted in Fig. 5, is a *NASOR* feature that allows establishing a multi-domain path, according to parameters defined by users. The interface implementations take into account the Graph data structure that *NASOR* abstracts from each domain topology. Therefore, this data structure holds nodes as routers and edges as links and allows them to assign properties to each one.

It also allows the implementation of the mechanism for searching paths, observing and assigning weights at the edges, defining nodes, and characteristics of links. In this sense, the domain administrator is responsible for managing the catalog of path definition policies for the OPI network slices.

Since *NASOR* considers that the policy is not the default, as shown on the left side of the illustration, it requests the home domain topology and forwards it to the OPI implementation. The alternative path choice mechanism receives an object containing the Graph data structure, which semantically represents the topology, and operates according to its path choice implementation.

Thus, the agent *NANO*, when receiving the Graph data structure processed through the mechanism, can configure the parameters of the network slice through remote procedure calls.

3.4. The multi-domain data plane

We refer to ASs as administrative domains from the routing perspective. Thus, the routing algorithms, precisely the inter-AS, are responsible for acting on the backbone linking the ASs. In the Internet routing area, especially in Internet eXchange Point (IXP), internal route announcements are observed. In the IXP area, *NASOR* chooses, according to Border Gateway Protocol (BGP) routes, a better datapath for the network slice between ASs or Internet Service Providers (ISPs). These announcements are subsidies for the routing algorithm operation, which provides a list of the best paths, taking into account the lowest cost metric, which considers the number of hops in a BGP path.

The inter-AS BGP routing algorithm establishes a control mechanism that behaves on two router structures. The exchanging route announcements occur systematically, according to the BGP, and allow the BGP session to build a Routing Information Base (RIB) structure. Additionally, the BGP control plane builds the Forwarding Information Base (FIB) structure and stores current and best cost information for forwarding a packet to a given route. The FIB structure has an instruction set for each packet that allows routing one interface instead of another, considering the best route computed by the routing algorithm.

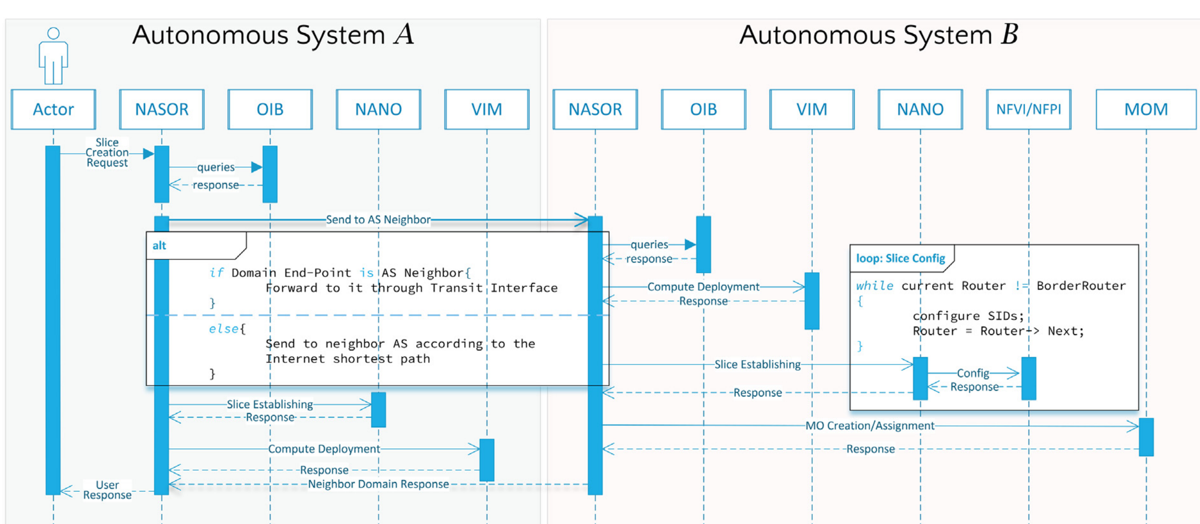

Fig. 4. Network slice deployment spanning across multiple domains.

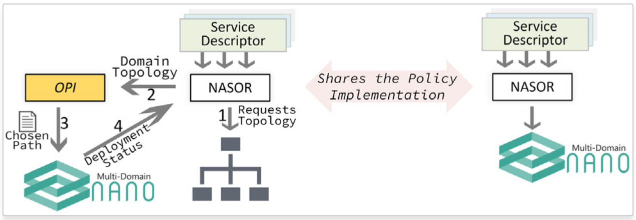

Fig. 5. Open Policy Interface (OPI) Schema.

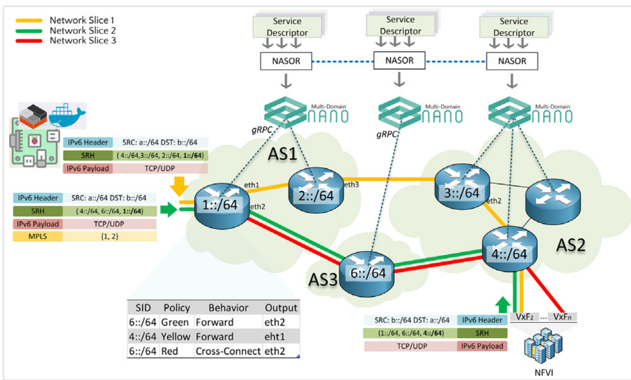

Fig. 6. *NASOR* Data Plane Approach.

Fig. 6 depicts the positioning and interaction of control and routing entities in network slice establishment. Our approach takes into account the routing data plane, which carries an additional forwarding instruction mechanism leveraged by segment routing, namely the Segment Identifier (SID) table and Segment Routing Header (SRH). As depicted, the *NANO* installs and changes the SIDs on tables and instructs the ingress route to properly encapsulate the SRH. All instructions to routers pass through *Google* Remote Procedure Call (*g*RPC) routines.

The control over the SID table represents the network slice management. When a slice change occurs, it must propagate the changes and the parameters to another AS. This control mechanism introduced by *NASOR* provides the building and maintenance of logical paths between router entities, eventually in different domains. In this respect, we argue that a network slice is the data and control plane of each logical path configured through segment routing parameters on top of Internet routers.

Additionally, Fig. 6 depicts a multi-domain scenario with three network slices installed over three different ASs. The network slicing for similar logic paths employs the coloring approach to differentiate the logical paths. Therefore, the same logical path can receive a different policy from its peers through the coloring parameter.

The *NASOR* data plane approach has access to alternative routes, unlike BGP or Open Shortest Path First (OSPF) low-cost paths. Therefore, it is possible to establish a network slice with an alternative path, eventually taking into account other metrics to interconnect the involved ASs. The RIB lookup in the routers, which is held through a remote procedure call, brings to the *NASOR* alternative routing interfaces that allow deploying an alternative logical path through the SIDs table and SRH encapsulation.

3.5. Mapping NASOR to standard frameworks

To make our vision of network slicing clear, we propose and describe *NASOR* from two viewpoints: first, for the Network Slicing deployment within Network Services for no 3rd Generation Partnership Project (3GPP) ecosystems such as Management and Orchestration

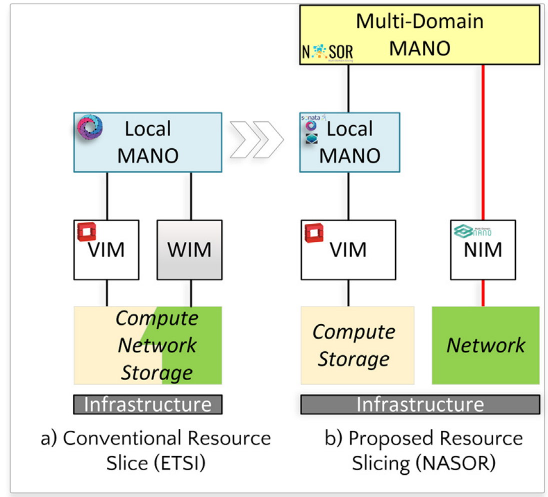

Fig. 7. The Network Slicing Management Vision extended from ETSI.

(MANO); second, for the Network Slices deployment to 3GPP, playing roles for network slicing realization in multi-domain Transport Networks (TN). The *NASOR* vision complies with these two perspectives, proposing a framework for the network slicing deployment over multiple ISP domains.

According to ETSI, a Network Service is a composition of organized and interconnected network functions. A Network Service may eventually contain one or more instances of Network Slices and interconnected physical or virtual network functions. *NASOR* plays networking roles in place of MANO once it does not fully address multi-domain scenarios, as shown in Fig. 7.

Alternatively, 3GPP recognizes that Network Slices supports Communication Services. Besides, from the perspective of 3GPP, Network Slice may contain Network Slice Subnets Instances (NSSIs) and Network Functions (NFs). Both in 3GPP and ETSI, there are interchangeable relationships and roles between the entities implementing network slicing. However, these entities and their roles need to be distinguished according to ETSI and 3GPP network slicing vision.

3.5.1. ETSI

As in SDN, in which there are data and control plane separations, other decouplings integrate the architectural frameworks of computational and network resources, such as the management and service plane. Within the scope of management and orchestration solutions, it is possible to assign roles to MANO and classify them as an entity in the management plane.

Therefore, the network slicing proposed by state-of-the-art solutions, as seen in Fig. 7, is coupled to the structure of the MANOs that provide inter-domain connectivity according to its WAN Infrastructure Manager (WIM) technology. Alternatively, according to Fig. 7b, the *NASOR* proposal organizes the MANOs in local and multi-domain, forking the management of inter-domain network slices from the Local MANO to Multi-Domain MANO through *NASOR* entity.

This approach allows separate management, increasing flexibility, programmability, and fine-grained requirements specification through Network Infrastructure Manager (NIM). It also leaves the Local MANO to handle the deployment of services within the data center domain. The Multi-Domain MANO uses East/Westbound interfaces to exchange deployment requests to other domains.

Hence, the network slicing assumption of *NASOR* relies on logical connectivity, established hop-by-hop, taking into account the Internet data plane, where each network slice contains management, control, and private data plane.

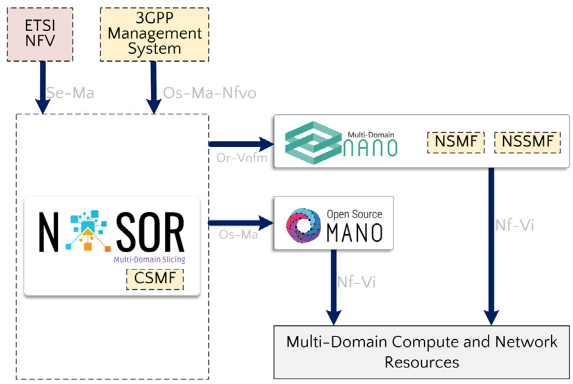

Fig. 8. Mapping 3GPP network slicing roles on *NASOR* entities.

The management plane refers to the components of handling the network and monitoring service instance. The control plane refers to the private entity that sets the parameters in network resources, namely routers, to deploy the logical connectivity. The data plane refers to the logical connectivity through the segment routing table, in which their entries semantics in each router allow establishing the connectivity.

3.5.2. 3GPP

As seen in Fig. 8, we mapped 3GPP entities and their roles against the *NASOR* framework to sheds light on general tasks. 3GPP specifies entities and roles to manage network slices, such as the Communication Service Management Function (CSMF), the entity responsible for translating the high-level communication specification into the slice instance specification.

There is no definition of implementing CSMF's role in the specification, especially for network slices deployed over a Transport Network (TN) spanning multiple ASs. Besides, *NASOR* receives a Communication Service descriptor through the Os-Ma-Nfvo interface describing a network slice between multiple domains and provides the implementation of network slices in the context of Transport Networks (TN) between multiple domains. Thus, from the perspective of 3GPP, NASOR implements and extends the CSMF role by proposing a distributed-CSMF to deploy multi-AS slices.

Additionally, *NASOR* is the entity that receives the Communication Service specification and interprets it to instantiate slices between multiple ASs. Thus, the 3GPP specification provides the entity Network Slice Management Function (NSMF) to deal with a network slice's life-cycle. Besides, 3GPP provides the entity Network Slice Subnet Management Function (NSSMF), which plays the role of managing network slices' sub-instances.

The *NASOR* framework encompasses and simultaneously implements both entities and roles within the NANO entity through the Or-Vnfm interface. Also, NANO enables deployment and management of multi-domain and recursive network slicing, that is, one network slice within the other managed independently. Additionally, *NASOR* provides management, control, and independent data planes for network slice instances deployed across multiple ASs.

4. Experimental scenario

As depicted in Fig. 9, we have two domains (ASs) communicating through a peering interface and advertising reachable network routes according to their internal policy. The domain 1 has VNFs clients connected directly to router $R1$ and router $R5$ containing VNF server in domain 2. Each AS has a specific internal routing area, which implies that edge routers need to advertise internal routes in the BGP session. Furthermore, it implies that the mechanism for defining the path for the network slice must take into account the OSPF and BGP path simultaneously.

Additionally, it is possible to observe three stripes, dashed black, red and green, that connect the routers along a path. The dashed line refers to the connectivity between the routers for advertising routes and transporting data. The others, green and red, and eventually others, refer to logical connectivity over the IPv6 data plane, created for VNFs communication. The green network slice, established by *NASOR*, takes into account the data path between the client VNF A and B chosen by the OSPF and BGP routing algorithms. The red network slice represents a defined path chosen through a third-party pathfinder, which runs on top of OPI.

Besides, there is a SID table on the router $R3$, which contains instructions installed for the two network slices instantiated for the proposed experiment. Each entry in the SID table contains a behavior, configured by the NANO Agent of the network slice, which instructs the packets that match the SID to go through a particular path adopt a defined behavior. The routers, comprising τ and λ domains, are managed by their home *NASORs* and communicate with each other to exchange information about the logical connectivity process.

The experiment dynamics are the *NASOR* of Domain 1 (AS1) when receiving the service descriptor file based on YANG containing the compute, and the network slice specification must process them immediately. In particular, the network slice specification contains information regarding the policy type used by the slice deployer. Hence, the path configuration takes into account that information to configure SIDs in order to create logical connectivity among multiple domains. Regarding the experiments, we chose OSM to handle computing requirements and implemented Voice over Internet Protocol (VoIP) VNF with two entities communicating by voice, client A and B. The transport of the VoIP packets crosses the ASs and experiences varying network conditions.

In the scenario of Fig. 9, we investigated two policies for establishing network slices between multiple domains. Thus, different paths between those domains that consider granular network metrics on each router are open to being defined by third party-application. At this point, the OPI (see Section 3.3) interface advances the state-of-the-art by allowing third approaches in path establishment without intervening in the political and technological independence of the domains.

Thus, the proposed experiment considers two policies for establishing logical connectivity between multiple domains—one based on the network and the other based on the data path provided by BGP and OSPF. We also added simulated link failures in order to assess the applicability of one policy over another. As shown in Fig. 9, a quality degrader is placed between the $R1$ and $R3$ routers, which should be taken into account in the determination of the path to the network slice. Thus, the proposal brings dynamism to the establishment of network slices.

In order to explore the OPI, we propose an algorithm to choose paths for establishing network slices over multiple domains. The Algorithm 1 explores the applicability of OPI. Each domain runs an instance of Algorithm 1 , where each one manipulates the data planes in the SID Table only in its domain. Upon finding the edge router that complies with the path choice policy, it has been assumed the next domain instance will configure the slice parameters by completing the end-to-end path.

The algorithm results are the choice of a path that considers, according to a greedy approach, the path that best satisfies the network quality policy within the AS to the edge router. As output, it brings a graph data structure that supports the NANO about the routers and interface parameters in order to provide the logical separation of the network slice.

We implemented the experimental scenario Fig. 9 on virtual machines connected through Open vSwtich (OVS). We preserve and configure the interfaces and their logical connections as depicted in the topology. Each virtual machine's flavor ran Debian 6.0 in *"squeeze"* release on top of 2 GB of RAM and 2 vCPU. Each virtual machine aimed to simulate an Internet router providing Intra and Inter-AS connectivity for clients and route announcements. We used the Quagga routing software suite [49] to run on each virtual machine.

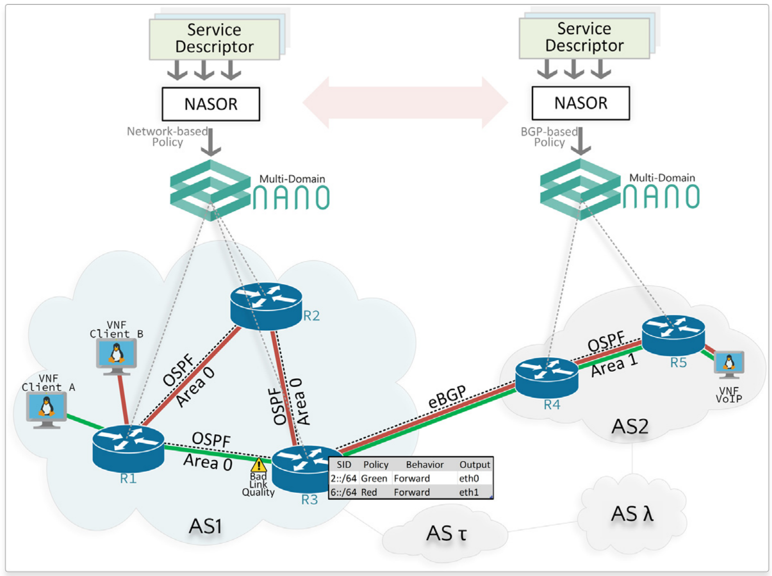

Fig. 9. Experimental Scenario: Path Definition Policies for Network Slices.

Algorithm 1: Multidomain Network Performance-based Path (MDPP).

Result: $\forall R \in$ {Paths between an Source s and a Destination d} $\rightarrow$ Choose a path that considers the smallest overhead location of the interfaces according to the policy described in the NSTD.
Input: A Targeted Graph that represents the Internal topology of the *NASOR* Domain received the request to establish the network slice

Consider *AS-PATH* from BGP for an end-to-end network slice and send deployment request to all *NASORs* of *AS-PATH*;
Choose the edge router that has the best performance peering/transit interface according to the metric;
Consider the Graph that has paths between s and d: compute a path according to the greedy strategy that best satisfies the metric to the edge router;
while *R is not end-point target* **do**
    if *R* == *Border Router* **then**
        Set up the *SIDs* and the Policies in the peering/transit interface;
        break;
    end
    Configure the table SID and Policies on current domain according to the path based on the network metric;
    $R = R \rightarrow Neighbor$;
end
Return to NANO the tree that represents the path chosen for the network slice;

4.1. Validation method

Regarding the previous scenario, we considered the time spent by *NASORs* for configuring the network slice across multiple domains. To this end, we recorded the instant time in a slicing, establishing request, and the completion slice parameter time to observe the elapsed time. Quantitatively, it is possible to understand the asymptotic behavior influence in the path definition algorithm on the network slice configuration time across the domains.

The time spent evaluation in the network slice establishment compares the proposal that takes into account the data path offered by the routing algorithms, designed as a baseline, with the other algorithms built by third-parties, which they consider as weights at the edge conditions of the network link.

Some Key Performance Indicators (KPIs) referring to slice run-time and life-cycle management of slices are known in the network slicing specification and support the performance evaluation [50,51]. Includes those KPIs: mobility, end-to-end delay, resource utilization, to name a few. In this paper, we consider the KPI Slice Deployment Time (SDT) of every single domain, realizing the sum of all the time spent to measure the total multi-domain deployment time. Also, we take into account the Integrity KPI [51], which relates to the ability of a network slice to deliver information end-to-end, considering the delay and throughput metrics experienced by an application.

The choosing the best path approach that befalls internally in the domains, considering the network policy parameters: latency between the links, allows without losing generality to describe the proposed Algorithm 1 as a *Dijkstra* network quality-based. The weights that are assigned to the edges respect this function: $y = f(x)$, where $f(x) = \overline{X}$, which is the average latency measured by the probe process between pairs of routers belonging to a candidate path. We create an RPC API that brings link measurements through the *vnstat* tool.

The network experience measurement on the network slices considered the jitter metric. We build one network slice, considering the data path proposed by OSPF and BGP, while the other held the network policy orientation approach. Voluntarily, in the OSPF data path, despite the smallest cost from the routing perspective, we inserted a quality deflector mechanism in the link in order to simulate the packet losses, latency, and jitter.

We performed it through the NetEm tool, which allows adding adverse conditions to a network link. It was selected due to the excellent accuracy given to the input parameters and the behavioral response of the network [52]. The deflator parameters arbitrarily used to degrade the link quality between the $R1$ and $R3$ routers are the following: 20 ms latency and 30 ms jitter occurring according to a normal distribution and with a correlation of 20% between the current package and the previous ones.

Regarding Integrity KPI, to assess the scalability of the *NASOR* on network slices deployment between multiple domains, we led a benchmark on our prototype considering the topology of Fig. 9. We evaluate both according to a partial factorial experimental. We measure the performance in a network slice using the *iperf3* tool [53], considering a scenario with two factors, two levels, and throughput and latency as response variables, as shown in Table 2.

Table 2
Methodology for the partial factorial project.

		Levels
Factors	A: Flavor	**1:** 1 vCPU & 1 Gb RAM **−1:** 2 vCPU & 2 Gb RAM
	B: *n.* Slices SID Table	**1:** 300 *k* Slices **−1:** 1 Slice

The equation which gives the partial factorial experimental design is: $y = q_0 + q_A x_A + q_B x_B + q_{AB} x_{AB}$, where their quotients are calculated according Eqs. (1), (2), (3), (4).

$$q_0 = \frac{1}{4} \times (y_1 + y_2 + y_3 + y_4) \tag{1}$$

$$q_A = \frac{1}{4} \times (-y_1 + y_2 - y_3 + y_4) \tag{2}$$

$$q_B = \frac{1}{4} \times (-y_1 - y_2 + y_3 + y_4) \tag{3}$$

$$q_{AB} = \frac{1}{4} \times (y_1 - y_2 - y_3 + y_4) \tag{4}$$

Following, the *Total Sum of Squares (TSS)* given by Eq. (5) will give the total variation of the response variables and the variations due to the influence of factor A, factor B and the interaction between A and B.

$$TSS = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 \tag{5}$$

The experiment aimed to answer the following questions: 1 – how much time does NASOR spent to deploy a network slice using the conventional approach which obeys the data plane path provided by the routing algorithms, compared to an approach that explores the Open Policy Interface, to define a path for network slice on the multi-domain scenario, according to Fig. 9? 2 – is there a scenario where the slice data path built taking into account the routing algorithms does not offer satisfactory performance compared to a network slice that deployed over an alternative path, one that considers the quality of the links? 3 – Regarding scalability, what is the most influential factor in the performance of a multi-domain network slicing, considering both throughput and latency?

5. Results

In order to answer the guiding questions, we ran 50 deployment requests against *NASOR* of Domain B, which follows the routing algorithm-built data path. Similarly, the other 50 deployment requests considered the OPI, hurled against the *NASOR* of Domain A. The graph depicted in Fig. 10 allows inferring the network slice deployment time, which takes into account the link quality real-time analysis of about 12.27 ms average below a 5% error. In contrast, the baseline deployment, which takes into account the data path of the routing algorithms and does not perform any real-time analysis of link quality, was 1.07 ms, also subject to the 95% of the confidence interval.

Quantitatively, it is possible to discuss the disparity in the time taken to implement one approach over another. Although the objective of the experiment is to evaluate the OPI suitability as a mechanism to establish network slices dynamically and closer to user requirements, it is essential to consider some mathematical aspects that affected the performance of the two approaches under discussion. The first highlight is the approach that considers a policy based on the network quality, which presents itself as closer to the user requirements, with two components that affect the final implementation time. The first one is the evaluation time that the probe engine takes on the links among candidate paths. Hence, the network slice description file established that *NASOR* should evaluate the link quality in the 2 second time frame.

Different evaluation times could fill the network slice description file. However, longer evaluation time for the link would lead to a constant additional time $K \times n$, where K is the evaluation time-lapse,

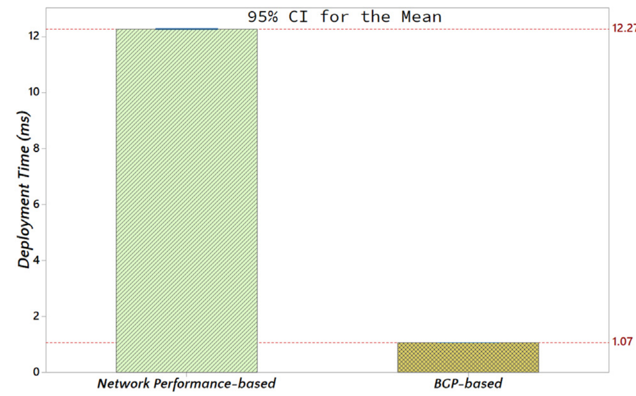

Fig. 10. Deployment Time considering Different Policies.

and n is the number of links to be evaluated. In this experiment, we compare the baseline time, which is the underlying mechanism of *NASOR*, namely: a mechanism for establishing network slices in multi-domain scenarios on top of the OSPF and BGP data plane against the network-aware approach. When exploiting the OPI and comparing the mechanisms for establishing paths supported by it with the baseline, the aim is to add and discuss the dynamism in network slicing operations. Additionally, we demonstrate that *NASOR* goes beyond the state-of-the-art by proposing dynamism in the establishment of inter-domain network slices.

The second aspect that impacts the network slice deployment is the nature of the path selection algorithm, a function that, given the input size, requires an amount of time to produce the output. For this experiment, which explores the OPI, two mathematical operations befall: (1) finding all paths given a source s and a destination d; (2) considering the link quality of those along each candidate path. Therefore, according to the algorithm complexity notation, the worst case to return the possible paths (in this experiment: candidate paths) is of the order $O(n!)$ [54]. Additionally, it should be noticed that it is an inefficient algorithm for n samples that are large enough.

In contrast, the network slicing deployment that considers the data path constructed by the routing algorithms does not experience this additional time overhead. When it is not oriented to the network policy, the network slicing establishing process passes through the elements and interfaces of the router control plane and configures the network slice parameters. The path computation time (reachable networks) is an essential task of the routing algorithms that do not affect the network slicing time by *NASOR*.

The *NASOR* mathematical formalism in sequential search process in a given router is polynomial n_{FIB_Search}. Thus, this operation is performed for each router n_{Router}, so $n_{Router} \times n_{FIB\_Search} \equiv n^2$, whose upper limit is $\mathcal{O}n^2$. Additionally, it is important to note the OSPF complexity time of the ASs, so we have $E \times \log V$, where E refers to links and V refers to routers [55,56].

Also, the Algorithm 1 is greedy-based. Nonetheless, a further efficient approach is acceptable, and according to [57], it is a *NP-hard* class problem whose merit transcends the objective of the experiment.

Slices management and orchestration mechanisms must be able to allocate resources considering efficiency. This allocation must be compatible and verifiable considering the two parts, the operator and the user [58]. In this sense, metrics of scalability, performance and manageability can support evaluating solutions for network slicing deployment. Another aspect of network slicing performance refers to the run-time [50], aiming this measurement, we consider a Voice over Internet Protocol (VoIP) application containing low-latency, higher throughput and jitter requirements, desired in enhanced Mobile Broadband (eMBB) and Ultra-reliable and Low-latency Communications (URLLC) applications [51,59–61].

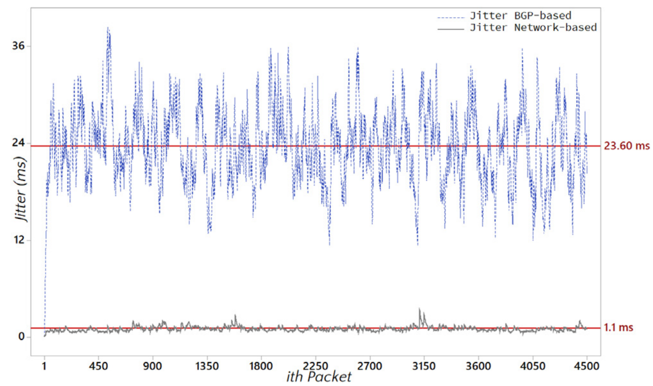

Fig. 11. Jitter experienced by the Voice over Internet Protocol (VoIP) application on network slices implemented on two types of path choices.

We carried out another experiment that assesses the impact of choosing a path for a network slicing procedure. Furthermore, it expands the previous one concerning application quality, and its results are reported in the graph of Fig. 11. According to this experiment, there are only two possible paths for establishing the network slice, which runs through routers $R1$, $R3$ and $R4$, and path $R1$, $R2$, $R3$ and $R4$. Additionally, from Fig. 11, it is possible to understand that there are two jitter quantities to be analyzed, an average of 23.60 *m* and another of 1.1 ms.

A VoIP packets flow containing about 4,500 UDP packets carrying voice chunks processed by the codec G711a [62] on top of created network slice. The observed jitter was 23.60 ms jitter in the green-colored network slice, according to the experimental scenario illustrated in Fig. 9. Moreover, the network slice is based on the routing algorithms, which has a quality degradation between the $R1$ and $R3$ routers voluntarily inserted.

The measured jitter on the red network slice was, on average, about 1.1 ms. The red slice was built according to the third-party mechanism, which uses the OPI, who measured a real-time link quality. After the algorithm assessing the link quality on the candidate paths and, as planned, the best path measured by the OPI mechanism is the one that runs the routers $R1$, $R2$, $R3$, and $R4$ consecutively. Thereby, the Voice over Internet Protocol (VoIP) application on that network slice experiences a jitter of 1.1 ms.

It is important to note these network slices, on which the Voice over Internet Protocol (VoIP) application carried its data, followed the same deployment steps as the previous experiment. Thus, the red network slice chosen after an exhaustive quality links evaluation in the two candidate paths. Also, the red network slice took more time, as already mentioned, because the link quality in the topology was systematically evaluated, namely: $R1 \leftrightarrow R2$, $R2 \leftrightarrow R3$, $R3 \leftrightarrow R4$, $R1 \leftrightarrow R3$ and $R3 \leftrightarrow R4$, respectively.

Thus, the OPI feature allows making some assertions. First, adding dynamism to the network slice establishment shed light in the state-of-the-art as a contribution. Considering the available solutions does not provide this dynamism, especially in the multi-domain modality. Secondly, depending on the purpose of the network slice, the data path based on the routing algorithms may not be the best decision.

Indeed, there are approaches to OSPF that balances traffic, multi-paths, and others that deal with scenarios such as equal weight for several possible paths. However, the most important aspect is the materialization of a mechanism for network slicing that allows the user/domain administrator to define the path of the network slice according to a policy described in the Network Slice Template Description (NSTD) file.

The NSTD file extends ETSI NST, including a new tag that specifies multiple ASs network slicing. Our NSTD file carries a description of the ASs, including the ASN source and destination and the application requirements specification, such as bandwidth and latency constraints, that will use the network slice.

Table 3
Experimental results.

Experiment	*Factors*		*Dependent variable*	
	vRouter Flavor	SID Table	$\overline{X}$ Latency (ms)	$\overline{X}$ Throughput (Mbps)
#1	−1	−1	0.608	990.6
#2	1	−1	0.489	1016
#3	−1	1	4.028	1162
#4	1	1	1.863	1204

Table 4
Influence of factors and their interaction.

Parameter	Latency (ms)		Throughput (Mbps)	
	Estimated average	Variation	Estimated average	Variation
q_0	1.747		1093.15	
q_A	−0.571	16.11%	16.85	3.39%
q_B	1.1985	**70.97%**	89.85	**96.40%**
q_{AB}	−0.5115	12.93%	4.15	0.21%
SST	8.096302		33496.67	

The results related to the *NASOR* scalability on the network slices deployment had been collected from the partial factorial experimental design. According to Table 2, Factorial Design is the structure of the performance evaluation method that we carried out in the NASOR evaluation. Hence, the experimental design containing the results with the sample means are according to Table 3, where each factor had been tried varying in two levels.

Thus, according to Table 4, the factor that most impacted the latency response variable is factor B, size of the SID table: in the experiment, it contained records of 300 thousand different slices, making a 70.97% influence on the response latency variable. Besides, it is possible to infer that factor A, the flavor of the virtual router, exerted 16.11% influence on the response variable followed by 12.93% influence factors A and B combined.

Furthermore, the results referring to the throughput response are available in Table 4. The Table allows us to infer that the factor that most influenced the throughput was the number of slices implanted, as shown by the number of entries in the SID table, implying 96.40% of the influence on throughput. Besides, factor A referring to the flavor of the virtual router exerted a 3.39% influence on the throughput performance experienced by the slice running *iperf3*, followed by the minimal influence of the combination of factors A and B, with about 0.21%.

6. Concluding remarks

This work presented a network slicing framework for multiple domains on top of the Internet data plane. Our approach, leveraged by SDN, NFV, and Segment Routing technologies, allowed the underlying domain infrastructures to apply configurations entity-by-entity and grant connectivity to the user applications.

Besides, it advanced the state-of-the-art, providing dynamism in establishing the multi-domain network slices and bringing a recursive network slicing framework. Carried experiments in simulated scenarios highlighted the potential of this proposal's feasibility and scalability to compose the current network infrastructure.

We have also consolidated a comparative table, classifying state-of-the-art approaches according to multi-domain features to demonstrate the technological and scientific outcomes (see Section 2.4). Additionally, we have positioned the applicability and advances of *NASOR* against its peers.

Besides, this paper sheds light on the current understanding of service orchestration, traditionally computer-based. The proposed framework describes essential entities for network slice management. It allows a network slice to be resized exhaustively while respecting the original slice's restrictions and capabilities.

Additionally, *NASOR* brought an interface that extends the essential management and orchestration functionality, thereby increasing the fine-grained network slice specification level in service customization. The OPI proposal aimed to consolidate dynamism in establishing network slices across multiple domains.

Experimental results showcased that the OPI applicability to advance the state-of-the-art with dynamism for network slices deployment. It is also possible to gather from the experiments that the additional overhead in the network slice deployment has succeeded with enhancements in Voice over Internet Protocol (VoIP) applications' Quality of Service.

It also exemplified the life-cycle management and run-time behavior of a network slice towards the eMBB requirements in a multi-domain scenario. Besides, a partial factorial experimental, considering the KPI Integrity, showcases that the slice table (SID) size is the most influential factor in the latency and throughput in a scalability scenario with several concurrent slices.

Some research challenges lay further. First, we need to evaluate new methods for guarantee and improving Quality of Service and reliability for network slices deployed through *NASOR*. Knowing these methods is relevant because they guarantee connectivity under controlled variations, maintaining application requirements. Second, it seems relevant and challenging to propose and evaluate intelligent multi-domain network slicing methods that consider the economics of resources and the requirements' satisfaction according to a service agreement. We consider it relevant for a network slice to offer a connectivity ecosystem without wasting resources. Lastly, security in network slicing is challenging, especially security in distributed architectures with open interfaces, which bring interoperability and security concerns [63].

CRediT authorship contribution statement

Rodrigo Moreira: Methodology, Software, Investigation, Writing – original draft. **Pedro Frosi Rosa:** Writing – review & editing. **Rui Luis Andrade Aguiar:** Visualization, Validation, Writing – review & editing, Conceptualization. **Flávio de Oliveira Silva:** Supervision, Investigation, Writing – review & editing, Conceptualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This project was partially funded under project 5GAIner, Brazil, a 5G+AI, Brazil joint university-industry initiative in Aveiro.

References

[1] Cisco, Internet adoption and network performance, 2018–2023, 2018, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf. (Accessed 03 2020).

[2] NGMN Alliance, Description of network slicing concept, 2016, https://www.ngmn.org/wp-content/uploads/Publications/2016/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf. (Accessed 07 2020).

[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, Network slicing and softwarization: A survey on principles, enabling technologies, and solutions, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2429–2453.

[4] Z. Yuan, G.M. Muntean, AirSlice: A network slicing framework for UAV communications, IEEE Commun. Mag. 58 (11) (2020) 62–68.

[5] A. Rostami, P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos, A. Vidal, Orchestration of RAN and transport networks for 5G: An SDN approach, IEEE Commun. Mag. 55 (4) (2017) 64–70.

[6] X. Zhou, R. Li, T. Chen, H. Zhang, Network slicing as a service: enabling enterprises' own software-defined cellular networks, IEEE Commun. Mag. 54 (7) (2016) 146–153.

[7] R. Ricart-Sanchez, P. Malagon, A. Matencio-Escolar, J.M.A. Calero, Q. Wang, Toward hardware-accelerated QoS-aware 5G network slicing based on data plane programmability, Trans. Emerg. Telecommun. Technol. 31 (1) (2020) e3726.

[8] N. Bastin, A. Bavier, J. Blaine, J. Chen, N. Krishnan, J. Mambretti, R. McGeer, R. Ricci, N. Watts, The InstaGENI initiative: An architecture for distributed systems and advanced programmable networks, Comput. Netw. 61 (2014) 24–38, Special issue on Future Internet Testbeds – Part I.

[9] F.S.D. Silva, M.O.O. Lemos, A. Medeiros, A.V. Neto, R. Pasquini, D. Moura, C. Rothenberg, L. Mamatas, S.L. Correa, K.V. Cardoso, C. Marcondes, A. ABelem, M. Nascimento, A. Galis, L. Contreras, J. Serrat, P. Papadimitriou, NECOS project: Towards lightweight slicing of cloud federated infrastructures, in: 2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft, 2018, pp. 406–414.

[10] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J.J. Ramos-Munoz, J. Lorca, J. Folgueira, Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges, IEEE Commun. Mag. 55 (5) (2017) 80–87.

[11] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Pérez, Network slicing games: Enabling customization in multi-tenant mobile networks, IEEE/ACM Trans. Netw. 27 (2) (2019) 662–675.

[12] R. Mahindra, M.A. Khojastepour, H. Zhang, S. Rangarajan, Radio Access Network sharing in cellular networks, in: 2013 21st IEEE International Conference on Network Protocols, ICNP, 2013, pp. 1–10.

[13] A. Basit, S. Qaisar, M. Ali, M. Naeem, M. Bruyere, J.J.P.C. Rodrigues, Interconnecting networks with optimized service provisioning, Telecommun. Syst. 73 (2) (2020) 223–239.

[14] M.L. Héno, A. Boubendir, N. Simoni, Telco network slicing models enabling recursive multi-tenancy, in: 2019 10th International Conference on Networks of the Future, NoF, 2019, pp. 40–47.

[15] T. Taleb, I. Afolabi, K. Samdanis, F.Z. Yousaf, On multi-domain network slicing orchestration architecture and federated resource control, IEEE Netw. 33 (5) (2019) 242–252.

[16] R. Moreira, P.F. Rosa, R.L.A. Aguiar, F. de O. Silva, Enabling multi-domain and end-to-end slice orchestration for virtualization everything functions (VxFs), in: L. Barolli, F. Amato, F. Moscato, T. Enokido, M. Takizawa (Eds.), Advanced Information Networking and Applications, Springer International Publishing, Cham, 2020, pp. 830–844.

[17] R. Moreira, F. de O. Silva, P.F. Rosa, R.L.A. Aguiar, A smart network and compute-aware Orchestrator to enhance QoS on cloud-based multimedia services, Int. J. Grid Util. Comput. 11 (1) (2020) 49–61.

[18] A. Császár, W. John, M. Kind, C. Meirosu, G. Pongrácz, D. Staessens, A. Takács, F. Westphal, Unifying cloud and carrier network: EU FP7 project UNIFY, in: 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, 2013, pp. 452–457.

[19] B. Sonkoly, J. Czentye, R. Szabo, D. Jocha, J. Elek, S. Sahhaf, W. Tavernier, F. Risso, Multi-domain service orchestration over networks and clouds: A unified approach, SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 377–378.

[20] ETSI, Open source MANO (OSM), project, 2016, OSM Home Page, https://osm.etsi.org/. (Accessed 07 2020).

[21] Linux Foundation, Open network automation platform (ONAP), 2017, https://www.onap.org/. (Accessed 09 2019).

[22] N.F.S. de Sousa, D.A.L. Perez, R.V. Rosa, M.A.S. Santos, C.E. Rothenberg, Network service orchestration: A survey, Comput. Commun. 142–143 (2019) 69–94.

[23] M. Series, IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond, 2015, Recommendation ITU, 2083, https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf. (Accessed 06 2020).

[24] ETSI, Network functions virtualisation (NFV) release 3; evolution and ecosystem; report on network slicing support with ETSI NFV architecture framework; GR NFV-EVE 012, 2017, https://www.etsi.org/deliver/etsi_gr/NFV-EVE/001_099/012/03.01.01_60/gr_NFV-EVE012v030101p.pdf. (Accessed 08 2020).

[25] S. Dräxler, H. Karl, M. Peuster, H.R. Kouchaksaraei, M. Bredel, J. Lessmann, T. Soenen, W. Tavernier, S. Mendel-Brin, G. Xilouris, SONATA: Service programming and orchestration for virtualized software networks, in: 2017 IEEE International Conference on Communications Workshops, ICC Workshops, 2017, pp. 973–978.

[26] X. Li, R. Casellas, G. Landi, A. de la Oliva, X. Costa-Perez, A. Garcia-Saavedra, T. Deiss, L. Cominardi, R. Vilalta, 5G-crosshaul network slicing: Enabling multi-tenancy in mobile transport networks, IEEE Commun. Mag. 55 (8) (2017) 128–137.

[27] A. Lutu, B. Jun, F.E. Bustamante, D. Perino, M. Bagnulo, C.G. Bontje, A first look at the IP EXchange ecosystem, SIGCOMM Comput. Commun. Rev. 50 (4) (2020) 25–34.

[28] C.J. Bernardos, B.P. Gerö, M.D. Girolamo, A. Kern, B. Martini, I. Vaishnavi, 5GEx: Realising a Europe-wide multi-domain framework for software-defined infrastructures, Trans. Emerg. Telecommun. Technol. 27 (9) (2016) 1271–1280.

[29] Z. Kotulski, T.W. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, J.P. Wary, Towards constructive approach to end-to-end slice isolation in 5G networks, EURASIP J. Inf. Secur. 2018 (1) (2018) 2.

[30] A. de la Oliva, X. Li, X. Costa-Perez, C.J. Bernardos, P. Bertin, P. Iovanna, T. Deiss, J. Mangues, A. Mourad, C. Casetti, J.E. Gonzalez, A. Azcorra, 5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals, IEEE Commun. Mag. 56 (8) (2018) 78–84.
[31] H. Koumaras, D. Tsolkas, G. Gardikis, P.M. Gomez, V. Frascolla, D. Triantafyllopoulou, M. Emmelmann, V. Koumaras, M.L.G. Osma, D. Munaretto, E. Atxutegi, J.S.d. Puga, O. Alay, A. Brunstrom, A.M.C. Bosneag, 5GENESIS: The genesis of a flexible 5G facility, in: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, 2018, pp. 1–6.
[32] M.A. Kourtis, T. Anagnostopoulos, S. Kukliński, M. Wierzbicki, A. Oikonomakis, G. Xilouris, I.P. Chochliouros, N. Yi, A. Kostopoulos, L. Tomaszewski, T. Sarlas, H. Koumaras, 5G network slicing enabling edge services, in: 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN, 2020, pp. 155–160.
[33] A.P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. de O. Silva, I. Vidal, B. Nogales, R. Nejabati, D. Simeonidou, 5GinFIRE: An end-to-end open5G vertical network function ecosystem, Ad Hoc Netw. 93 (2019) 101895.
[34] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S.K. Kasera, E. Lewis, D. Maas, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L.B. Stoller, J.V. der Merwe, K. Webb, G. Wong, POWDER: platform for open wireless data-driven experimental research, in: Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, WiNTECH'20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 17–24.
[35] M.U. Alhuseini, M.M. Olama, 5G service value chain and network slicing framework using ecosystem modeling, agile delivery, and user-story automation, IEEE Access 7 (2019) 110856–110873.
[36] T. Taleb, B. Mada, M. Corici, A. Nakao, H. Flinck, PERMIT: Network slicing for personalized 5G mobile telecommunications, IEEE Commun. Mag. 55 (5) (2017) 88–93.
[37] F.Z. Yousaf, V. Sciancalepore, M. Liebsch, X. Costa-Perez, MANOaaS: A multi-tenant NFV MANO for 5G network slices, IEEE Commun. Mag. 57 (5) (2019) 103–109.
[38] G. Sun, Y. Li, H. Yu, A.V. Vasilakos, X. Du, M. Guizani, Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks, Future Gener. Comput. Syst. 91 (2019) 347–360.
[39] T. Huang, F.R. Yu, C. Zhang, J. Liu, J. Zhang, Y. Liu, A survey on large-scale software defined networking (SDN) testbeds: Approaches and challenges, IEEE Commun. Surv. Tutor. 19 (2) (2017) 891–917.
[40] S. Zhang, An overview of network slicing for 5G, IEEE Wirel. Commun. 26 (3) (2019) 111–117.
[41] Q. Ye, J. Li, K. Qu, W. Zhuang, X.S. Shen, X. Li, End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework, IEEE Veh. Technol. Mag. 13 (2) (2018) 65–74.
[42] A. Csoma, B. Sonkoly, L. Csikor, F. Németh, A. Gulyas, W. Tavernier, S. Sahhaf, ESCAPE: Extensible service chain prototyping environment using mininet, click, NETCONF and POX, in: Proceedings of the 2014 ACM Conference on SIGCOMM, SIGCOMM '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 125–126.
[43] T. Saboorian, A. Xiang, L. Thiébaut, Network slicing and 3GPP service and systems aspects (SA) standard, 2017, IEEE Softwarization.
[44] ETSI, Network functions virtualisation; architectural framework, 2013, p. v1, ETSI GS NFV, 2, https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf. (Accessed 05 2020).
[45] C.K. Tham, Y. Liu, Assured end-to-end QoS through adaptive marking in multi-domain differentiated services networks, Comput. Commun. 28 (18) (2005) 2009–2019, Current Areas of Interest in End-to-End QoS.
[46] H.A. Mantar, A QoS routing architecture for multi-domain networks, in: 2006 International Symposium on Computer Networks, 2006, pp. 236–241.
[47] J. Xiao, R. Boutaba, QoS-aware service composition in large scale multi-domain networks, in: 2005 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005, IM 2005, 2005, pp. 397–410.
[48] J. Zhao, F. Li, D. Ren, J. Hu, Q. Yao, W. Li, An intelligent inter-domain routing scheme under the consideration of diffserv QoS and energy saving in multi-domain software-defined flexible optical networks, Opt. Commun. 366 (2016) 229–240.
[49] P. Jakma, D. Lamparter, Introduction to the quagga routing suite, IEEE Netw. 28 (2) (2014) 42–48.
[50] S. Kukliński, L. Tomaszewski, Key performance indicators for 5G network slicing, in: 2019 IEEE Conference on Network Softwarization, NetSoft, 2019, pp. 464–471.
[51] 3GPP, Management and Orchestration; 5G end to end Key Performance Indicators (KPI), 3rd Generation Partnership Project (3GPP), 2018, https://www.etsi.org/deliver/etsi_ts/128500_128599/128554/15.00.01_60/ts_128554v150001p.pdf. (Accessed 12 2020).
[52] K. Velásquez, E. Gamess, A comparative analysis of WAN emulators, in: Proceedings of the 7th Latin American Networking Conference, LANC '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 44–51.
[53] J. Dugan, S. Elliott, B.A. Mah, J. Poskanzer, K. Prabhu, *iperf3*, tool for active measurements of the maximum achievable bandwidth on IP networks, 2014, URL: https://github.com/esnet/iperf.
[54] R. Sedgewick, Algorithms in C, Part 5: Graph Algorithms, third ed., Addison-Wesley Professional, 2001.
[55] R. Doss, Future Network Systems and Security: Second International Conference, FNSS 2016, Paris, France, November 23–25, 2016, Proceedings, Springer, Cham, Switzerland, 2016.
[56] S. Dasgupta, Algorithms, McGraw-Hill Higher Education, Boston, 2008.
[57] L. Gong, Y. Wen, Z. Zhu, T. Lee, Toward profit-seeking virtual network embedding algorithm via global resource capacity, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 1–9.
[58] P. Rost, M. Breitbach, H. Roreger, B. Erman, C. Mannweiler, R. Miller, I. Viering, Customized industrial networks: Network slicing trial at hamburg seaport, IEEE Wirel. Commun. 25 (5) (2018) 48–55.
[59] G. Brown, Exploring 5G new radio: Use cases capabilities & timeline, Qualcomm White Paper, 2016, https://www.qualcomm.com/media/documents/files/heavy-reading-whitepaper-exploring-5g-new-radio-use-cases-capabilities-timeline.pdf. (Accessed 04 2020).
[60] ITU, Y. 1541: Network Performance Objectives for IP-Based Services, International Telecommunication Union, ITU-T, 2003.
[61] S. Abbas, M. Mosbah, A. Zemmari, ITU-T Recommendation G. 114,"One way transmission time, in: International Conference on Dynamics in Logistics 2007, LDIC 2007, in: Lect. Notes in Comp. Sciences, Citeseer, 1996.
[62] Y. Hiwasaki, H. Ohmuro, ITU-T G. 711.1: extending G.711 to higher-quality wideband speech, IEEE Commun. Mag. 47 (10) (2009) 110–116.
[63] J.M. Batalla, E. Andrukiewicz, G.P. Gomez, P. Sapiecha, C.X. Mavromoustakis, G. Mastorakis, J. Zurek, M. Imran, Security risk assessment for 5G networks: National perspective, IEEE Wirel. Commun. 27 (4) (2020) 16–22.

Rodrigo Moreira is Assistant Professor at the Federal University of Viçosa campus Rio Paranaíba and received his B.S. degree from the Federal University of Viçosa and his M.S. degree from the Federal University of Uberlândia, Brazil, in 2014 and 2017 respectively. He is a Ph.D. candidate in computer science at the Federal University of Uberlândia. His research interests include future internet, quality of service, cloud computing, network function virtualization, software-defined networking, and edge computing.

Pedro Frosi Rosa received the PhD degree in Electrical Engineering in 1995 from the University of São Paulo (USP). He is currently a Full Professor in the Faculty of Computing at the Federal University of Uberlândia (UFU), Minas Gerais – Brazil. He has experience in areas such as computer science and engineering, with emphasis on highly scalable and highly available architectures, cloud computing, software defined networking, network functions virtualisation, with more than 150 articles published.

Rui L. Aguiar received the PhD degree in Electrical Engineering in 2001 from the University of Aveiro, where he is currently a Professor. He is leading a research team at the Institute of Telecommunications, and is an invited researcher at Federal University of Uberlândia, Brazil. His current research interests are centered on the implementation of advanced networks and systems with special emphasis on future Internet and 5G architectures, and he is currently involved in the 5G-PPP initiative. He is a member of ACM, with more than 350 published papers. He has served as technical and general chair of several conferences, from IEEE, ACM and IFIP, and is regularly invited for keynotes on 5G and future internet networks. He is the current Chair of the steering board of the Networld2020 ETP.

Flávio de Oliveira Silva is a professor at the Faculty of Computing (FACOM) in the Federal University of Uberlândia (UFU) and received a Ph.D. degree in 2013 from the University of São Paulo. Member of ACM, IEEE, and SBC, he has several papers published and presented in conferences around the world. He is a reviewer for several journals and member of TCPs of several IEEE conferences. Future Networks, IoT, Network Softwarization (SDN and NFV), Future Intelligent Applications and Systems, Cloud Computing, and, Software based Innovation are among his main current research interests.