Enabling Multi-domain and End-to-End Slice Orchestration for Virtualization Everything Functions (VxFs)

Rodrigo Moreira[1,2(✉)], Pedro Frosi Rosa[1], Rui Luis Andrade Aguiar[3], and Flávio de Oliveira Silva[1]

[1] Faculty of Computing (FACOM), Federal University of Uberlândia, Uberlândia 38400-902, Brazil
{rodrigo.moreira,pfrosi,flavio}@ufu.br
[2] Institute of Exact and Technological Sciences, Federal University of Viçosa, Rio Paranaíba 38810-000, Brazil
rodrigo@ufv.br
[3] Telecommunications Institute (IT), University of Aveiro, Aveiro, Portugal
ruilaa@ua.pt

Abstract. The traditional business models exploited in networking are changing into industry verticals, which in turn builds new applications with striking and specific requirements. Therefore, the service-oriented, fully programmable, and flexible features that translate to sliced-capable networks are fundamentals in the design, deployment, and orchestration of networks such as 5G and beyond. Also, application consumption experiences are moving towards pervasiveness, and it is necessary to address the established inter-domain constraints uniformly. Leveraged by SDN, Cloud/Edge Computing, and NFV, several state-of-the-art proposals aim to address multi-domain slice deployment. However, they focus on multi-domain control plane efforts, leaving numerous data plane challenges openly. This paper seeks to overcome the multi-domain slice establishing issues through a source routing and BGP-based approach to provide slice abstraction to cope with application requirements. A proof-of-concept called NASOR was implemented and validated using *VxFs* use-cases. The results showcase its deployment suitability in traditional core networks and enhancement of the end-user experience.

Keywords: Multi-domain · NFV · Segment Routing · SDN · Network slicing

1 Introduction

The network Core needs new management design and formats. According to the Cisco forecast report, in 2022, 71% of IP traffic will be generated by mobile and wireless devices [1]. Also, Ericsson asserts that by 2025, there will be 8*bn* mobile broadband subscriptions, of which 2.6*bn* will be 5G. The data volume generated

L. Barolli et al. (Eds.): AINA 2020, AISC 1151, pp. 830–844, 2020.
https://doi.org/10.1007/978-3-030-44041-1_73

by these subscribers will be 160 exabytes, wherein 76% is video traffic [2]. Hence, fully-automated management and service orientation features are mandatory at various network levels. Therefore, efforts to address this challenge are discussed and developed by academia and industry.

Internet Service Providers (ISP) infrastructure comprises a myriad of devices from several vendors and specific domains and levels such as Core, WAN, Radio, and Edge. Also, the nature of the telecommunications market is fragmented, so there are many networks and cloud operators. The heterogeneity of domains and their technologies has brought with it the challenges of fully-management and resource orchestration. Also, the connectivity format of these infrastructures is complex, error-prone, expensive, and very time-consuming. Also, compute resources management, formerly separate from network resources management, has now been handled uniformly to take into account quality of service and programmability [3].

Through new technology enablers such as Software-Defined Networking (SDN) [4], Network Function Virtualization (NFV) [5], Cloud/Edge Computing [6], Source Routing (introduced by RFC 7855), specifically Segment Routing (SR) [7–10], the network slicing has taken shape in the 5G network specification and beyond. According to [11–13], the network slicing concerns an instance of the network, which has management independent of the physical layer and beyond as an essential feature to provide network slicing.

Advances in network Core are early stages, whereas RAN, Edge, and Access, there are significant advances [14–17]. Hence, we propose NASOR, a fully-layered, multi-domain slice establishing a solution for Virtualization Everything Functions *(VxFs)*. The NASOR underlying mechanism of multi-domain network slicing takes into account the behavior of the BGP control plane and SR capabilities for establishing and configuration of the data plane slice in a hop-by-hop way. It considers a distributed database for domain-specific announcements of computing capabilities. With this, NASOR performs network slicing and makes the connectivity between *VxFs* spanning across multi-domain feasible.

The main contributions of this paper are: (1) an extension of the ETSI method to deploy network services; (2) an iterative BGP-based process to establish multi-domain network slicing; (3) a distributed mechanism for exchange compute resources capabilities across multi-domains; (4) feasibility assessment of the Lightweight DNS as an edge-aware approach to handling essential network functions; (5) a state-of-the-art taxonomy for end-to-end multi-domain network slicing.

The remaining of the paper is organized as follows: In Sect. 2, we place our work in context of related work concerning multi-domain slicing establishing. In Sect. 3, we describe our solution and its features to cope with state-of-the-art challenges. Section 4 describes a proof-of-concept scenario and the evaluation methodology to asses our solution feasibility. Section 5 presents the results, following in Sect. 6 summarizes our findings, and we point out some research directions.

2 Related Work

Below we describe some related work that relies on multi-domain slicing establishing and orchestration, and we bring a taxonomy for classifying state-of-the-art approaches. There are numerous approaches that started the concept of resource slicing such as X-BONE [18], VIOLIN [19], VINI [20], FlowVisor [21], OFELIA [22] leveraged by enabler technologies mentioned above, networking slicing has taken shape in academia and industry.

The Extensible Service ChAin Prototyping Environment (ESCAPE) materializes the UNIFY [23] goals by using Mininet, Click, NETCONF, and POX. It has three intercommunication layers: Service Layer, Orchestrator Layer, which interprets the request and allocates it to resources. The Infrastructure Layer contains the resource management mechanism that carries out network, storage, and compute. The ESCAPE control plane reaches VIMs on different domains, but the data plane is limited to the SDN domain [24].

X-MANO Proposes a framework for deploying virtualized network services for administratively and technologically different domains. The proposal establishes that the principles of confidentiality of each domain should be maintained through the inter-domain data exchange interfaces. Scalability and political issues rest with centralized control, as it is necessary to define in which domain the central entity will be placed. Thus, the hierarchical approach may not be adequate when considering the performance and focal point of failure over the central entity. Thus, to address these issues, the NASOR introduces the concept of distributed end-to-end slice control over a carrier-grade data plane [25]. Similarly, XOS brings a cloud operating system that manages hardware and software resources in commodity clouds, private data centers, and clusters that form across the network [26].

The OpenFlow in Europe Linking Infrastructure and Applications (OFELIA) is a well-established approach to address experimenting challenges of new network services from a geographically distributed perspective, by the popularization of OpenFlow protocol [22]. Alternatively, 5GEx is a project proposal that envisions a multi-domain service orchestration ecosystem with hybrid virtualization technologies, inspired by the IPExchange concept [27]. Different from NASOR, both use the VPN or SDN-based networking.

The OSM is an ecosystem aligned with the ETSI specification for VNF life-cycle management. The OSM architecture contains three distinguished components: MANO, VNF, and NFVI. Different from NASOR, the OSM provides service connectivity through a cloud-native SDN-based data plane; therefore, it is not possible to establish a multi-domain slice [28]. Similarly, the SONATA architecture complies with the ETSI framework aims to provide a consistent and integrated solution for complete virtualized network service life-cycle management. Both solutions do not provide multi-domain slice establishing [29].

The NECOS project is based on the Lightweight Slice Defined Cloud (LSDC) concept and materializes a Slice-as-a-Service approach that spans over multiple cloud computing infrastructures. NECOS aims to address the challenges of deploying applications and services by network operators and service providers.

The inter-domain communication of NECOS takes into account the cloud network domain, which can be SDN-based. Also, the NECOS connectivity approach has gaps in to provide multi-domain network slicing [30].

A 5GinFIRE is a FIRE-based [31] ecosystem compliant with the ETSI NFV reference model to provide a playground for application experimenting. Also, 5GinFIRE has advanced the state-of-the-art testbeds, including the scalability of the ETSI MANO framework for defining and describing experiments and proposing the concept of the vertical industry in the 5G mainstream [32].

Table 1 organizes contributions according relevant state of the art outcomes. The Data-Plane Multi-Domain column is the ability to support data exchange across multi-domains. The Control-Plane Multi-Domain column refers to the ability to influence features and behaviors in different domains on service deployment. Controller Architecture characterizes the solution in terms of its core component organization and peer interaction. The Network Place column corresponds to the geographic location of the Orchestration main component. The End-To-End Slice refers to the applicability multi-domain network slicing to provide data-plane connectivity from edge to core and spanning across other domains.

The Inter-Domain Data Exchange highlights the enabling technologies to provide *VxF* connectivity. Deployment Place refers to the ability to deploy, from a customer perspective, Indoor, Outdoor, or Hybrid services. Indoor deployment befalls when service deployment occurs on top of domestic compute resources (e.g., Raspberry Pi), or malls, airports, or sports stadiums. When outdoor, they are deployed in base stations, back-haul, or core network. Legacy Network Compliant refers to the ability to be embodied in an ISP without significant infrastructure and protocols modifications. Finally, the MANO Compliant column proposes to define the solutions that fit the ETSI framework.

Table 1. State-of-the-art approaches comparison.

Solution	Data-Plane Multi-Domain	Control-Plane Multi-Domain	Controller Architecture	Network Place	End-to-End Slice	Inter-Domain Data Exchange	Enabling Technologies	Deployment Place	Legacy Network Compliant	MANO Compliant
ESCAPE [24]	No	Yes	Monolitic	Simulation	No	Directly (SDN Domain)	SDN + Cloud	Outdoor	No	No
X-MANO [25]	Yes	Yes	Hierarchical	Core	No	IP Tunnel	Cloud	Outdoor	No	No
XOS [26]	Yes	Yes	Monolitic	Core	No	VLAN, Internet	SDN + Cloud	Outdoor	Yes	No
OFELIA [22]	No	No	Monolitc	Data Center	No	VPN	SDN + Cloud	Outdoor	No	No
OSM [28]	No	No	Monolitic	Data Center	No	SDN Domain or VPN	SDN + Cloud	Outdoor	No	Yes
5GEx [27]	Yes	Yes	Distributed	Core	No	Internet	SDN + Cloud	Outdoor	Yes	Yes
NECOS [30]	No	No	Distributed	Data Center + Edge	No	VLAN + SDN	SDN + Cloud + Container	Indoor + Outdoor	No	Yes
SONATA [29]	No	Yes	Monolitic	Data Center	No	SDN Domain or VPN	SDN + Cloud + Container	Indoor + Outdoor	Yes	Yes
5GinFIRE [32]	No	No	Centralized	Data Center	No	VPN	SDN + Cloud	Outdoor	No	Yes
NASOR	Yes	Yes	Distributed	Core + WAN + Edge	Yes	IPExchange	SDN + Cloud + Container	Indoor + Outdoor	Yes	Yes

The state-of-the-art proposals aim to deal with multi-domain slice establishing, and there are numerous well-established commitments in the control plane [33]. However, the data plane uses techniques known and traditionally oriented to single domains (SDN, VLAN, and others).

3 Network and Slice Orchestrator (NASOR): Architecture

To address open issues regarding the deployment of *VxFs* on a multi-domain scenario, we propose the NASOR[1] solution. The architecture is three-layered, which brings scalability, customization, and domain independence. It comprises the layered architecture sketch, as shown in Fig. 1, the Edge, WAN, and Core Controllers.

Components managed by Edge Controller communicate with each other asynchronously. Slice Mapper operates as Mux/Demux of slices, binding a container to a virtual interface, and instructs the interface to encapsulate in Service Slice Identifier (SSI) packages for differentiation between slices. Slice information is persisted in distributed Domain Information Base (DIB) repositories. They are subdivided into: inter-domain, where information shared across domains, such as Network and Orchestrator (NANO) agents, is persisted. Also, intra-domain refers to technologically and politically unique information from each domain, such as routing policy and ingress and egress entities.

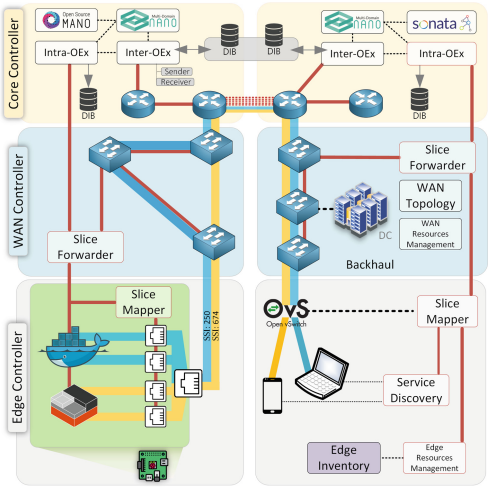


Fig. 1. NASOR architecture – comprising: edge controller, WAN controller, and core controller

The Service Discovery agent presents to users, upon request, with a set of service slices (such as CDN, lightweight DNS servers) available for deployment and use at the edge. We consider as a service slices containers which are logically interconnected and virtual machines in a multi-domain perspective that perform defined functions. According to [32], we refer to the concept of virtualized network functions as virtualization of any service namely *VxF*.

[1] Artifacts available at https://github.com/romoreira/EdgeComputingSlice.

Edge Inventory agent, through two approaches, monitors edge capabilities. The resource management approach, by a probe, deals with hardware features such as memory, CPU, and Storage. Catalog management handles hardware characteristics as owners, roles, and deployed service slices. The NASOR solution is capable of deploying *VxF* in indoor (user home, mall, and others) and outdoor (service provider cloud and MECs) domains over bare-metal hardware or ARM.

Next is the WAN Controller, which contains the Slice Forwarder, WAN Topology, and WAN Resources Management modules. The Slice Forwarder module establishes a data plane between WAN elements that provides Edge to Core domain connectivity, and this logical connectivity happens through the label-switched path (LSP) through an MPLS network. The labels used in this approach are persisted in the DIB of each domain. WAN Topology, through the neighborhood protocol, maintains state and connectivity between entities. The WAN Resources Management module identifies the hardware state (ports and capacity) of L2 entities to handle SLA aspects described in the NSTD.

At the topmost layer is the Core Controller, through its agents ensures the persistence of slices across routing devices and between neighboring domains. Inspired by the inter-network operation of the Internet, the NASOR solution ensures that edge entities from each domain are being connected according to on-boarded service policy. Core network entities use the concept of segment routing to distinguish and route slice traffic properly. SID policies are used to identify slices on inter-domain data-plane.

The Intra-Orchestrator Exchange (Intra-OEx) and Inter-Orchestrator Exchange (Inter-OEx) modules forwards slice deployment requests between underlying controllers and to neighboring domain core controllers. Also, these modules directly access the two DIB levels: intra-domain and inter-domain. The first contains information regarding the routing algorithm, topology information, ingress and egress entities between network levels, and available target MANOs. The second one persists information shared across domains, NANO agents, and service deployment capability matching.

3.1 Edge Agent

Within three Python 3.7 scripts, Edge Agent performs its functionalities. They run alongside Docker and LXD container managers. We adapted the container creation and network port assignment process; as a result, the container interface is attached to Open vSwitch (OvS) on launching time. Slice Mapper queries the domain repository to choose which SSI will encapsulate the packets from the container. As proof-of-concept, we used MPLS labels as in [34].

The Service Discovery performs a query in the intra-DIB repository and presents the service slices available on the domain. A Service Discovery query made from another location may eventually bring other services. The Edge Inventory implementation is a script that reads from *top* command the Raspian system statistics and publishes them to the domain repository. We built the DIB repository with the Apache Geode mainly because of its data storage format (JSON and key-value).

3.2 WAN Controller

The WAN Controller is an SDN application that runs on top of the Ryu controller. We choose Ryu as a build-block solution because it is open-source, performs well, and it has campus domain compatibility [35,36]. The WAN Controller defines an LSP from the ingress switch to the egress switch, where it goes to the core domain. Packets from containers are labeled when they enter the WAN domain, and they are treated according to the MPLS label.

The WAN Controller topology engine is an extension of the Ryu Topology application, which is based on LLDP. Also, viewing entities and their connections is possible by the extension made on the NetworkX package. WAN Controller uses the topology to define LSPs for service slices and to apply QoS policies. The resource manager receives sFlow (of each switch) statistics and persists it on DIB.

3.3 Core Controller

This controller comprises three main components: *NANO*, a listener application that triggers asynchronous messages to underlying controllers, and parallel domain controllers in order to deploy a multi-domain slice. Messages are carried by the second component, the *OrchestratorExchange*, which communicates with domain controllers and neighboring NANO controllers. The message content contains the service descriptors (ETSI-based), the request identifier, and the origin (ASN).

Algorithm 1 describes the NASOR procedures to build multi-domain network slicing establishing. The algorithm output is: for every Router that belongs to the shortest-path computed by the BGP control plane, establishing an especial data-plane which we call network slicing. Interactively, it checks if the current Router is the slice target, and it configures the SIDs, and Policies (colors) or forwards to the next Router.

The third component of the Core Controller build-block is the managers of *VIM plugins*. The NASOR framework carries Open Source Mano (OSM) compatibilities for deploying *VxFs* on bare-metal infrastructures. Also, the Core Controller uses xmlrpc-based API of Open Nebula Edge 5.8 to provide container deployment as a service slice on edge. In the Core Controller, the control plane communicates with Quagga-based routers through South-Bound Interface (SBI), a gRPC-based customized from [37]. NANO uses that interface to install SID as part of the slice configuration in the onboarding process. Another gRPC-based SBI has been developed that watches topology modifications to change the assignment of SIDs and policies.

Algorithm 1. Multi-Domain Network Slicing Establishing (MD-NSE).

Result: $\forall$ $R \in$ {BGP Shortest–Path} $\rightarrow$ Multi-Domain Network Slicing Establishing
GET from DIB Router specifications and Slice details from NSTD;
while *R is not slice target* **do**
  Configure in shortest Path: SIDs and Policies from DIB and NSTD file;
  if $R ==$ *Border Router* **then**
    Forward NSTD to neighboring NANO;
  else
    $R = R \rightarrow Neighbor$;
  end
end

4 Evaluation

To assess the NASOR suitability to provide *VxFs* deployment in multi-domains, we propose the Lightweight DNS use case. As depicted in Fig. 2, the scenario explores its ability to ensure the cross-domain slice establishing of *VxFs* and the applicability of edge computing as enabling technology to decrease application latency. NASOR receives via NBI the VNFD, NSD, and NSTD files containing *VxF* MANO-compliant specifications as well as the multi-domain slice descriptor that includes the ASs involved in the slice. For the experimental scenario, OSM 5 is used, which manages an OpenStack-based VIM with previously configured images. We built the *VxF* Lightweight DNS on top of an Ubuntu-16 image with the *Bind9* application to run as bare-metal VM and container-based.

Figure 2 depicts three domains (ASs) wherein the BGPv6 routing algorithm establishes the data plane. Each router embodies the SR implementation that enables it to assigns SIDs and policies to the shortest path interfaces and services. On top of the illustration, the NANO entity receives the manifest files carrying the service descriptors and forwards it to the Core Controller and OSM. Thus,

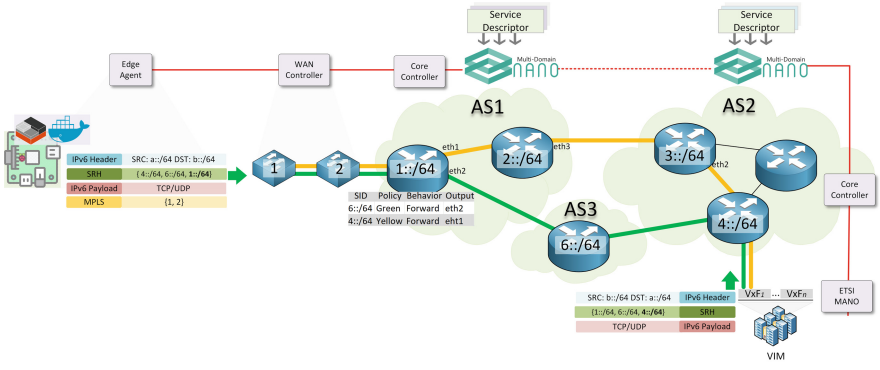


Fig. 2. Experimental setup of inter-domain end-to-end slice establishing.

based on Algorithm 1 queries the DIB and NSTD to install SIDs and policies on interfaces take into account the shortest path measured by the BGP control plane.

Before *VxFs* inject packets targeting its destination, they add a header to the package; the Segment Routing Header (SRH) comprises the predefined path to span slice across multi-domain networks. In establishing a slice process, upon finding the domain boundary, the information enclosed in the manifest files defining the service reaches to the NANO entity of the other domain. When the slice creation is complete, the Core Controller forwards to the WAN Controller (in the same domain) the specification to properly setup switches in order to materialize slice on the underlying layer. Lastly, it enables Edge Agent to configure container interfaces on edge to properly direct packets to the ingress/egress slice interface.

4.1 Evaluation Methodology

In order to validate NASOR, we propose two experiments: the first one highlights the suitability of our multi-domain slicing approach in order to improve the end-user perception of DNS lookup response time. To this end, we measure the latency of DNS lookup perceived by end-users in multi-domain established slice by using the Namebench tool [38]. The second, we measure the end-to-end latency overhead caused by the processing of multi-domain slice capable packets (SRH) using the *Flent* tool [39]. We verify the performance of our multi-domain network slicing approach against the non-slice capable baseline scenario and VPN based. All experiments considered a network without congestion and extraneous data.

Similar to the methodology of [40], we evaluated the performance of *VxF* LW-DNS described and deployed according to the ETSI specification on multi-domain infrastructure. The *VxF* LW-DNS is hosted on a Raspberry PI 3 and receives DNS queries by the *VxF* Client through the multi-domain established slice. Mostly, home users use the DNS service provided by their ISP [40], alternatively with *VxF* LW-DNS we propose new findings regarding edge-based DNS against to Google DNS and OpenDNS.

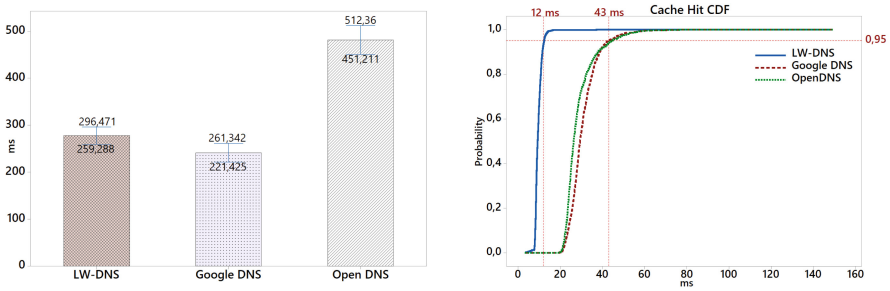
Studies such as [41,42] introduced and evaluated the benefits of DNS closer to end-users, however, we brought the concept of Local DNS to an edge computing use-case. To avoid biases in the responsiveness metric of the DNS service, we consider from the Alexa [43] base the 2,000 most popular and 2,000 least popular queries. The main goal is to answer the following questions: Is it statistically better to use high-performance DNS infrastructures rather than edge computing approaches? What is the time overhead that packages experience in multi-domain network slicing (established through the NASOR approach)?

5 Results

From the graph depicted in Fig. 3a, we can infer there is no statistical performance advantage by using DNS infrastructures such as Google DNS or OpenDNS

against LW-DNS. The response time of a DNS lookup, ignoring cache hit, of *VxF* LW-DNS compared to Google DNS although higher than average, considering the 95% confidence interval are statistically equal.

Additionally, in Fig. 3b the cache hit scenario shows by the accumulated probability graph that 95% of the cache hit queries will provide the end-user a response experience of up to ≈ 12 ms whereas Google DNS and Open DNS greater than ≈ 43 ms, i.e. a response time ≈ 3 times longer. According to the study [44], most user DNS lookup is known, so user perception can be improved. Therefore, it is reasonable to question why high-performance DNS approaches such as Google DNS or OpenDNS are seen in scenarios with security and privacy constraints [45–47].



(a) LW-DNS: Querry.

(b) LW-DNS Cache Hit.

Fig. 3. Measurement performance for multi-domain network slicing.

The processing overhead of introducing NASOR as a multi-domain slice approach as illustrated in Fig. 4 is ≈ 6%. Thus, it is 95% certainty that the latency experienced by a *VxF* on a multi-domain slice is between 1.59 ms and 1.70 ms. Traditional FlowVisor or VPN-based network slicing approach (e.g. OFELIA [22] or 5GinFIRE [32]), albeit in pretty different scenarios, experience significantly higher latencies than the scale in question [21,48,49]. For the experiment, a *VxF* client and server logically connected via VPN experience latencies from 6.45 ms to 7.40 ms. The baseline value is the end-to-end delay experienced by two *VxF* deployed in distinct domains without any inherent network slicing peculiarity as per the scenario in Fig. 2.

There is no predefined path that takes packets from a domain A to domain B, so the baseline latency refers exclusively to routing IPv6 packets as the least cost path. On the other hand, the overhead in the slice-based approach occurs through the processing of the passing SRH packet and the routing to the lowest-cost path computed by the BGP control plane. In the baseline scenario, it is not possible to establish quality metrics as they are subject to the multi-domain best-effort policy. In contrast, the NASOR approach makes it possible to apply QoS policies to multi-domain established slices.

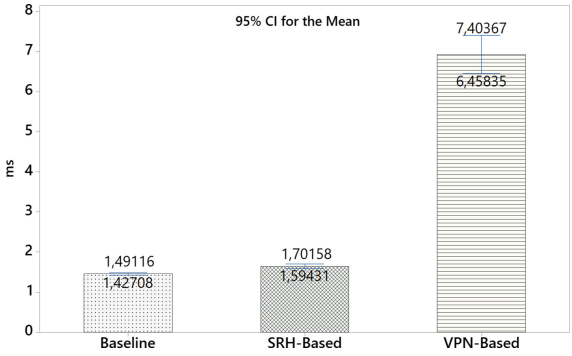


Fig. 4. Multi-domain network slicing overhead by using SRH.

6 Concluding Remarks

In this work, we propose and implement a multi-layered approach for multi-domain networking slicing establishing based on the BGP control plane named NASOR.

Through a taxonomy, we highlight the potentialities of our proposal against its peers. We demonstrate NASOR's advances in making inter-domain *VxF* deployment complaint with ETSI framework through SR. We built and validated the proposal for a distributed database (DIB) that persistently maintained technological and political aspects of domains, as well as information from inter-domain computing resources. Finally, we leverage our solution as a holistic and end-to-end approach to providing multi-domain slicing establishing.

Additionally, we showcase that as state-of-the-art points out, edge-based *VxFs* deployed closer to end-user could enhance its experience, especially for latency-sensitive applications. Also, we exploit the capabilities of NASOR to provide deployment in indoor and outdoor scenarios, opening the range of solutions of this nature. Also, edge-based DNS proposal affords considerable performance against traditional ones, as well as could leverage the level of security and privacy. Furthermore, we do not assert that it is always statistically better to use high-performance DNS approaches against edge-based. Our findings have shown the additional overhead on the SRH-based slicing proposal. Also, we conclude that it could not significantly compromise the performance of the service slices.

For future work, we are working on a forwarding plane considering technologies such as DPDK and VPP to address Ultra-Reliable Low-Latency Communication (URLLC) slices requirement. Also, we propose to evaluate the performance of *VxF* slices in a network with distinct traffic.

Acknowledgements. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

1. Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017–2022 (2019). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html. Accessed 26 Nov 2019
2. Ericsson: Ericsson Mobility Report (2019). https://www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf. Accessed 26 Nov 2019
3. Moreira, R., de Oliveira Silva, F., Rosa, P.F., Aguiar, R.: A flexible network and compute-aware orchestrator to enhance QoS in NFV-based multimedia services. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 512–519, May 2018
4. Nunes, B.A.A., Mendonca, M., Nguyen, X.-N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. **16**(3), 1617–1634 (2014)
5. ETSI, N.F.V.: Architectural framework. ETSI GS NFV **2**, v1 (2013)
6. Satyanarayanan, M.: The emergence of edge computing. Computer **50**, 30–39 (2017)
7. Abdullah, Z.N., Ahmad, I., Hussain, I.: Segment routing in software defined networks: a survey. IEEE Commun. Surv. Tutor. **21**, 464–486 (2019). Firstquarter
8. Filsfils, C., Nainar, N.K., Pignataro, C., Cardona, J.C., Francois, P.: The segment routing architecture. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, December 2015
9. Davoli, L., Veltri, L., Ventre, P.L., Siracusano, G., Salsano, S.: Traffic engineering with segment routing: SDN-based architectural design and open source implementation. In: 2015 Fourth European Workshop on Software Defined Networks, pp. 111–112, September 2015
10. Previdi, S., Horneffer, M., Litkowski, S., Filsfils, C., Decraene, B., Shakir, R.: Source packet routing in networking (SPRING) problem statement and requirements (2016)
11. IETF: Packet Network Slicing using Segment Routing (2019). https://tools.ietf.org/html/draft-peng-lsr-network-slicing-00. Accessed 26 Nov 2019
12. European Telecommunications Standards Institute (ETSI): Next Generation Protocols(NGP); E2E Network Slicing Reference Framework and Information Model (2019). https://www.etsi.org/deliver/etsi_gr/NGP/001_099/011/01.01.01_60/gr_NGP011v010101p.pdf. Accessed 26 Nov 2019
13. Next Generation Mobile Networks (NGMN): Description of Network Slicing Concept (2019). https://www.ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf. Accessed 26 Nov 2019
14. Ksentini, A., Nikaein, N.: Toward enforcing network slicing on RAN: flexibility and resources abstraction. IEEE Commun. Mag. **55**, 102–108 (2017)
15. Fernandez, J.-M., Vidal, I., Valera, F.: Enabling the orchestration of IoT slices through edge and cloud microservice platforms. Sensors **19**(13), 2980 (2019)
16. Lee, Y.L., Loo, J., Chuah, T.C., Wang, L.: Dynamic network slicing for multitenant heterogeneous cloud radio access networks. IEEE Trans. Wirel. Commun. **17**, 2146–2161 (2018)
17. Meneses, F., Fernandes, M., Corujo, D., Aguiar, R.: SliMANO: an expandable framework for the management and orchestration of end-to-end network slices. In: IEEE International Conference on Cloud Networking - CloudNet, November 2019

18. Touch, J.D.: X-bone. Technical report. University of Southern California Marina Del Rey Information Sciences Inst (2003)
19. Jiang, X., Xu, D.: Violin: virtual internetworking on overlay infrastructure. In: Cao, J., Yang, L.T., Guo, M., Lau, F. (eds.) Parallel and Distributed Processing and Applications, pp. 937–946. Springer, Heidelberg (2005)
20. Bavier, A., Feamster, N., Huang, M., Peterson, L., Rexford, J.: In vini veritas: realistic and controlled network experimentation. SIGCOMM Comput. Commun. Rev. **36**, 3–14 (2006)
21. Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N., Parulkar, G.: Flowvisor: a network virtualization layer. OpenFlow Switch Consortium, Technical report, vol. 1, p. 132 (2009)
22. Suñé, M., Bergesio, L., Woesner, H., Rothe, T., Köpsel, A., Colle, D., Puype, B., Simeonidou, D., Nejabati, R., Channegowda, M., Kind, M., Dietz, T., Autenrieth, A., Kotronis, V., Salvadori, E., Salsano, S., Körner, M., Sharma, S.: Design and implementation of the OFELIA FP7 facility: the European openflow testbed. Comput. Netw. **61**, 132–150 (2014). Special issue on Future Internet Testbeds - Part I
23. Császár, A., John, W., Kind, M., Meirosu, C., Pongrácz, G., Staessens, D., Takács, A., Westphal, F.: Unifying cloud and carrier network: Eu fp7 project unify. In: 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, pp. 452–457, December 2013
24. Sonkoly, B., Czentye, J., Szabo, R., Jocha, D., Elek, J., Sahhaf, S., Tavernier, W., Risso, F.: Multi-domain service orchestration over networks and clouds: a unified approach. SIGCOMM Comput. Commun. Rev. **45**, 377–378 (2015)
25. Francescon, A., Baggio, G., Fedrizzi, R., Orsini, E., Riggio, R.: X-MANO: an open-source platform for cross-domain management and orchestration. In: 2017 IEEE Conference on Network Softwarization (NetSoft), pp. 1–6, July 2017
26. Peterson, L., Baker, S., De Leenheer, M., Bavier, A., Bhatia, S., Wawrzoniak, M., Nelson, J., Hartman, J.: XoS: an extensible cloud operating system. In: Proceedings of the 2nd International Workshop on Software-Defined Ecosystems, BigSystem 2015, pp. 23–30. ACM, New York (2015)
27. Bernardos, C.J., Gerö, B.P., Di Girolamo, M., Kern, A., Martini, B., Vaishnavi, I.: 5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures. Trans. Emerg. Telecommun. Technol. **27**(9), 1271–1280 (2016)
28. ETSI, O.: Open source mano. OSM home page (2016)
29. Dräxler, S., Karl, H., Peuster, M., Kouchaksaraei, H.R., Bredel, M., Lessmann, J., Soenen, T., Tavernier, W., Mendel-Brin, S., Xilouris, G.: Sonata: service programming and orchestration for virtualized software networks. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 973–978, May 2017
30. Silva, F.S.D., Lemos, M.O.O., Medeiros, A., Neto, A.V., Pasquini, R., Moura, D., Rothenberg, C., Mamatas, L., Correa, S.L., Cardoso, K.V., Marcondes, C., ABelem, A., Nascimento, M., Galis, A., Contreras, L., Serrat, J., Papadimitriou, P.: Necos project: towards lightweight slicing of cloud federated infrastructures. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 406–414, June 2018
31. Gavras, A., Karila, A., Fdida, S., May, M., Potts, M.: Future internet research and experimentation: the fire initiative. SIGCOMM Comput. Commun. Rev. **37**, 89–92 (2007)

32. Silva, A.P., Tranoris, C., Denazis, S., Sargento, S., Pereira, J., Luís, M., Moreira, R., Silva, F., Vidal, I., Nogales, B., Nejabati, R., Simeonidou, D.: 5GinFIRE: an end-to-end open5G vertical network function ecosystem. Ad Hoc Netw. **93**, 101895 (2019)
33. Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A.: 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Comput. Netw. **167**, 106984 (2020)
34. Gifre, L., Ruiz, M., Velasco, L.: CASTOR: a monitoring and data analytics architecture to support autonomic domain and slice networking. In: 2018 20th International Conference on Transparent Optical Networks (ICTON), pp. 1–4, July 2018
35. Salman, O., Elhajj, I.H., Kayssi, A., Chehab, A.: SDN controllers: a comparative study. In: 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1–6, April 2016
36. Stancu, A.L., Halunga, S., Vulpe, A., Suciu, G., Fratu, O., Popovici, E.C.: A comparison between several software defined networking controllers. In: 2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), pp. 223–226, October 2015
37. Ventre, P.L., Tajiki, M.M., Salsano, S., Filsfils, C.: SDN architecture and southbound APIs for IPV6 segment routing enabled wide area networks. IEEE Trans. Netw. Serv. Manag. **15**, 1378–1392 (2018)
38. Google, Namebench: Open-source DNS Benchmark Utility (2019). https://code.google.com/archive/p/namebench/. Accessed 25 Nov 2019
39. Høiland-Jørgensen, T., Grazia, C.A., Hurtig, P., Brunstrom, A.: Flent: the flexible network tester. In: Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2017, pp. 120–125. ACM, New York (2017)
40. Ager, B., Mühlbauer, W., Smaragdakis, G., Uhlig, S.: Comparing DNS resolvers in the wild. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement IMC 2010, pp. 15–21. ACM, New York (2010)
41. Mao, Z.M., Cranor, C.D., Douglis, F., Rabinovich, M., Spatscheck, O., Wang, J.: A precise and efficient evaluation of the proximity between web clients and their local DNS servers. In: USENIX Annual Technical Conference, General Track, pp. 229–242 (2002)
42. Shaikh, A., Tewari, R., Agrawal, M.: On the effectiveness of DNS-based server selection. In: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), vol. 3, pp. 1801–1810, April 2001
43. Alexa top sites (2019). http://www.alexa.com/topsites. Accessed 25 Nov 2019
44. Qian, H., Rabinovich, M., Al-Qudah, Z.: Bringing local DNS servers close to their clients. In: 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, pp. 1–6, December 2011
45. Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., Somaiya, N.: T-DNS: connection-oriented DNS to improve privacy and security (poster abstract). SIGCOMM Comput. Commun. Rev. **44**, 379–380 (2014)
46. Shulman, H.: Pretty bad privacy: pitfalls of DNS encryption. In: Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES 2014, pp. 191–200. ACM, New York (2014)
47. Nakatsuka, Y., Paverd, A., Tsudik, G.: PDoT. In: Proceedings of the 35th Annual Computer Security Applications Conference on - ACSAC 2019 (2019)

48. Chen, J., Ma, Y., Kuo, H., Hung, W.: Enterprise visor: a software-defined enterprise network resource management engine. In: 2014 IEEE/SICE International Symposium on System Integration, pp. 381–384, December 2014
49. Anvari, M., Broderick, T., Stein, H., Chapman, T., Ghodoussi, M., Birch, D.W., Mckinley, C., Trudeau, P., Dutta, S., Goldsmith, C.H.: The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery. Comput. Aided Surg. **10**(2), 93–99 (2005)