Breaking Down Network Slicing: Hierarchical Orchestration of End-to-End Networks

Joao F. Santos* Wei Liu†, Xianjun Jiao†, Natal. V. Neto¶, Sofie Pollin‡, Johann M. Marquez-Barja§, Ingrid Moerman†, and Luiz A. DaSilva*
* CONNECT - Trinity College Dublin, Ireland, {facocalj, dasilval}@tcd.ie
† IDLab - Ghent University - imec, Belgium, {wei.liu, xianjun.jiao, ingrid.moerman}@ugent.be
‡ Department of Electrical Engineering - KU Leuven, Belgium, sofie.pollin@esat.kuleuven.be
§ IDLab - University of Antwerp - imec, Belgium, johann.marquez-barja@uantwerpen.be
¶ School of Computing - Federal University of Uberlandia, Brazil, natalneto@ufu.br

***Abstract*—Network slicing is one of the key enabling techniques for 5G, allowing Network Providers (NPs) to support services with diverging requirements on top of their physical infrastructure. In this paper, we address the limited support and oversimplified resource allocation on different network segments of existing End-to-End (E2E) orchestration solutions. We propose a hierarchical orchestration scheme for E2E networks, breaking down the E2E resource management and network slicing problems per network segment. We introduce a higher-level orchestrator, the hyperstrator, to coordinate the distributed orchestrators and deploy Network Slices (NSs) across multiple network segments. We developed a prototype implementation of the hyperstrator and validated our hierarchical orchestration concept with two proof-of-concept experiments, showing the NS deployment and the impact of the resource allocation per network segment on the performance of NSs. The results show that the distributed nature of our orchestration architecture introduces negligible overhead for provisioning NSs in our particular setting, and confirm the need of a hyperstrator for coordinating network segments and ensuring consistent QoS for NSs.**

***Index Terms*—Network Slicing, Network Orchestration, End-to-End Networks, Virtualisation, Distributed Intelligence**

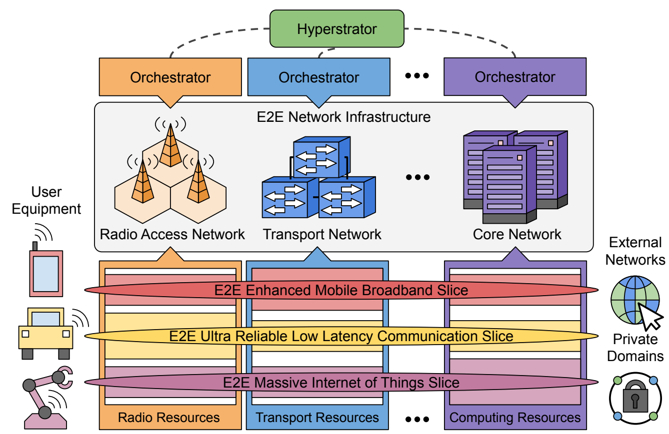

Fig. 1: Our proposed E2E network design with one specialised orchestrator per network segment, and a hyperstrator for coordinating the resource allocation across segments.

I. Introduction

In contrast to previous generations of mobile networks, 5G is envisioned from the very beginning to support a variety of different services, e.g., enhanced Vehicular-to-Everything (V2X), massive Internet of Things (IoT), and industrial automation [1]. To cope with the diverse requirements of such communication services, 3GPP introduced the concept of network slicing, which proposes partitioning the physical network infrastructure of Network Providers (NPs) into independent logical networks, known as Network Slices (NSs). Each NS operates as separate virtual networks, individually tailored and configured for serving different purposes, enabling NPs to simplify network management by assigning services with diverging requirements to different NSs [2].

The End-to-End (E2E) network infrastructure of NPs may comprise multiple network segments, each of which can be independently orchestrated, sliced, and combined for creating different types of NSs [3]. For mobile networks, the 3GPP defines NSs as the combination of Core Network (CN) slices and Radio Access Network (RAN) slices [4]. In other kinds of E2E networks, e.g., metro networks from Internet Service Providers (ISPs) or cloud networks from Cloud Providers, the NSs may consist of a combination of other types of Network Segment Slices (NSSs), e.g., Transport Network (TN) and Data Centre Network (DCN) slices [5]. Every NSS contributes to the overall QoS of an NS, as the inappropriate resource allocation in a single NSS not only leads to localised bottlenecks but also impairs the performance of all communication services running on the NS [6]. Therefore, guaranteeing consistent QoS for NSs requires cohesive resource allocation across multiple network segments [3]–[5].

Each type of network segment has different paradigms and abstractions, entailing distinct orchestration approaches that require domain expertise [2]. Consequently, there are specialised orchestrators, tailored for the particularities of specific segments, e.g., ONOS and OSM, providing fine-grained resource allocation in their respective segments. However, the interaction between different orchestrators for a joint orchestration of E2E networks remains an open challenge [5]. Conversely, there are one-size-fits-all orchestrators, which aim at orchestrating entire E2E networks, e.g., CORD [7] and 5G-EmPOWER [8], providing a central point of management for the entire E2E infrastructure. However, one-size-fits-all orchestrators tend to oversimplify the particularities of certain

	Orchestration Approaches			Supported Technologies				Network Segment Slicing			
	Control Centralisation	Decision Centralisation	Logic Disaggregation	LTE	WiFi	SDN	NFV	RAN	TN	CN	DCN
CORD [7]	Distributed	Centralised	Internal	C-RAN		✓	✓	✓	✓	✓	
5G-EmPOWER [8]	Distributed	Centralised	External	D-RAN	D-RAN	✓	✓	✓	✓		
Kista Orchestrator [9]	Distributed	Centralised	External	C-RAN		✓	✓	✓	✓		✓
ETSI NFVO-C [10]	Hierarchical	Distributed	Internal				✓			✓	✓
Hyperstrator [this work]	Hierarchical	Hierarchical	Internal	C/D-RAN	D-RAN	✓	✓	✓	✓	✓	✓

TABLE I: Qualitative comparison of characteristics and features present in existing E2E network orchestrators.

segments and support limited sets of technologies and standards [11], being exceedingly complex for adding new functionality, e.g., the state-of-the-art on resource management or incorporating new types of network segments, e.g., mmWave and satellite networks.

In this paper, we present the vision of the Orchestration and Reconfiguration Control Architecture (ORCA) Horizon 2020 project, and propose a hierarchical orchestration architecture for E2E networks. Our proposal addresses the oversimplified resource allocation and limited support for network segments of existing E2E orchestration solutions, by leveraging domain expertise and enabling the independent management of each segment, using distributed specialised orchestrators, as shown in Fig. 1. We introduce a higher-level orchestrator, the hyperstrator, to coordinate the distributed orchestrators and deploy NSs across multiple network segments. To the best of our knowledge, this is the first solution that decentralises both the control over the physical infrastructure and the decision over the resource management for E2E networks. These approaches facilitate upgrading or replacing underlying orchestrators, and including new segments or types of resources unforeseen at design time. Without loss of generality, in this work we focus on NSs comprised of RAN, TN, and CN segments in the administrative domain of a single NP.

In the following sections, we assess current one-size-fits-all orchestrators regarding their characteristics and features, as well as introduce our hierarchical orchestration scheme for E2E networks, leveraging the distributed intelligence of specialised orchestrators. Next, we detail the challenges and design choices for realising a prototype of our hierarchical orchestration scheme, as well as validate its ability to deploy customised NSs. Finally, we make our concluding remarks and pose some key open challenges on the hierarchical orchestration of E2E networks.

II. Current Approaches for E2E Orchestration

In the literature, there are many number of open-source initiatives that provide one-size-fits-all solutions for managing E2E networks. However, their orchestration approaches may differ regarding: the degree of centralisation of their control and decision functionality, i.e., centralised or distributed; the level of disaggregation of their orchestration logic, i.e., implemented by the orchestrator itself or by external applications; and their support for different administrative domains, network segments and technologies. In the remainder of this section, we review some notable examples of one-size-fits-all orchestrators and compare their different approaches to orchestration. Finally, by analysing of these solutions, we identify missing gaps in the state of the art on E2E network orchestration.

CORD aims at transforming the functionality of the central offices of NPs, from traditional facilities that provide legacy services using purpose-built hardware, into agile DCNs following the Everything as a Service (XaaS) paradigm [7]. CORD is popular among ISPs and Mobile Network Operators, supporting the instantiation of enterprise Content Delivery Networks and VPNs, as well as Baseband Units (BBUs) and Evolved Packet Cores (EPCs). CORD realises E2E communication services as chains of functions, using Network Function Virtualisation (NFV) to achieve fast and flexible deployment, and Software-defined Network (SDN) to connect Physical Network Functions (PNFs), e.g., physical servers and switches; and Virtual Network Functions (VNFs), e.g., virtual machines and containers [7]. However, CORD possesses limited capabilities regarding wireless network segments, only supporting Centralised-RAN (C-RAN) LTE deployments.

Similar projects, e.g., 5G-EmPOWER [8] and the Kista orchestrator [9], also leverage NFV and SDN to realise E2E communication services as chains of functions on shared commodity infrastructure. The former, 5G-EmPOWER, focuses on orchestrating E2E networks with multiple Radio Access Technologies (RATs). It is compatible with a wider variety of wireless network segments, supporting Distributed-RAN (D-RAN) deployments of LTE and Wi-Fi RATs, realised on Software-defined Radios (SDRs) and embedded Linux devices, respectively [8]. In contrast, the Kista orchestrator focuses on orchestrating E2E networks using C-RAN and different types of TN segments. It supports optical-based TNs for the fronthaul between BBUs and Remote Radio Heads, and packet-based TNs for the backhaul between BBUs and data centres [9]. Both 5G-EmPOWER and the Kista orchestrator disaggregate the orchestration logic from their platforms, not implementing any resource management directives, i.e., the intelligence behind the resource allocation and function placement. Instead, these solutions expect network applications, i.e., custom-made or third-party plugins, to manage the entire E2E network infrastructure. This approach facilitates NPs to programmatically define the behaviour of their networks, at the cost of requiring further development before initial usage.

The previous one-size-fits-all orchestrators interface with the underlying network infrastructure of a single administrative domain through distributed controllers. Each controller is responsible for carrying out resource allocation and function placement tasks on a specific type of network segment, simplifying the interaction between the orchestrator with heterogeneous hardware platforms [5]. However, these orchestrators take the opposite stance regarding their decision functionality, i.e., the intelligence behind the resource negotiation and management for the entire E2E network, centralising it in a single monolithic entity. This approach leads to complex and tightly integrated implementations, which makes including new functionality and supporting new types of network segments cumbersome and non-trivial [11]. Based on these issues, the conceptual work of [11] proposed a hierarchical orchestration architecture for coordinating NFV-based DCNs across multiple administrative domains. This paradigm was later standardised by ETSI [10], which proposed a higher-level NFV orchestrator, the NFVO-C, for decomposing service requirements and coordinating distributed lower-level entities, the NFVO-Ns, responsible for the resource management in their own data centres. However, these solutions only focus on orchestrating DCNs, not supporting managing the resources or slicing any type of wireless network segments. Table I summarises the results of our investigation, considering different orchestrator approaches, supported technologies, and slicing of network segments.

Public and private networks are evolving and incorporating new types of wireless network segments for serving current and future use cases, e.g., the addition of mmWave links, and the expected inclusion of satellite links for ubiquitous connectivity beyond 5G. NPs will need to integrate these new segments with their existing network deployments, and efficiently coordinate the use of heterogeneous resources across their extended E2E infrastructure. However, we observe that existing one-size-fits-all orchestrators are not suitable for such scenarios, due to their ossified centralised E2E network management, and the limited support for wireless network segments, with a coarse-grained placement of radio functionality that may lead to suboptimal E2E performance [12].

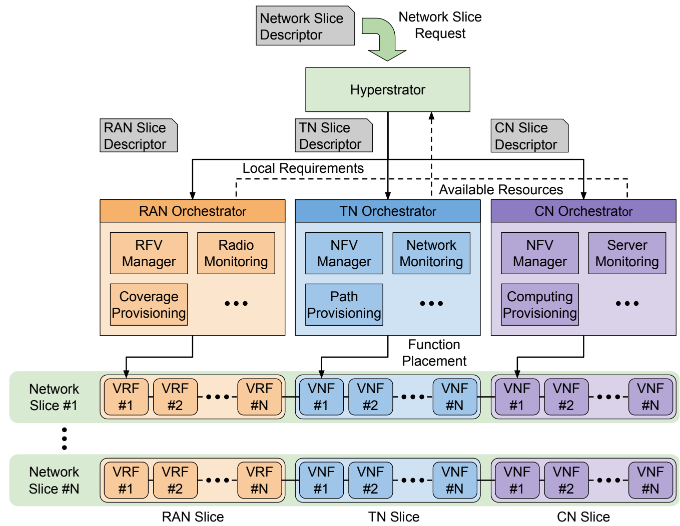

Fig. 2: Our hierarchical orchestration architecture, where each network segment has its own specialised orchestrator, while a new entity, the hyperstrator, accomplishes cross-network segment orchestration.

III. Hierarchical Orchestration of E2E Networks

In this section, we propose a hierarchical orchestration scheme for E2E networks, using a set of distributed orchestrators to manage different network segments, as shown in Fig. 2. Each orchestrator is responsible for the resource management in a particular segment, while we coordinate the resource allocation and functional placement across orchestrators and their respective segments through a higher-level orchestrator, namely, a hyperstrator. Our hierarchical orchestration architecture enables the decentralisation of the control and decision over E2E networks, breaking down the E2E resource management and network slicing problems into smaller, tractable problems per segment. However, it still provides a central point of management for the E2E network infrastructure, and hence, combines the benefits from both specialised and one-size-fits-all orchestration solutions.

In the remainder of this section, we introduce the hyperstrator and detail how it leverages multiple specialised orchestrators for managing E2E networks, while ensuring cohesive E2E resource allocation across network segments.

A. Coordinating Distributed Network Orchestrators

Creating NSs imposes both local requirements for specific network segments, e.g., coverage areas on RAN segments, or points of exchange on TN segments; and global requirements for the entire E2E network, e.g., throughput, delay, and reliability targets that the combination of network segments must meet to comply with the QoS of the NSs. Due to the different purposes and independence between segments, we can separate the E2E resource management per segment as long as we ensure that each segment delivers the necessary performance for meeting the global requirements [3] [5] [10]. We leverage this separation to decentralise the E2E network orchestration, delegating the decision over the resource allocation and function placement for each network segment to a respective specialised orchestrator. This allows each individual orchestrator to focus on a limited number of well-defined tasks, using paradigms and abstractions tailored for the particularities of its respective segment. This facilitates fine-grained resource allocation in every network segment, while reducing the resource management complexity, both in terms of design and implementation.

This decentralised orchestration paradigm requires coordination between different orchestrators for deploying NSs, a process that involves dimensioning, creating, and combining multiple NSSs. The distributed orchestrators could interact amongst themselves through east-westbound interfaces following a Self Organising Network (SON) model. However, this would require each orchestrator to implement new communication interfaces; be aware of the existence and capabilities of other orchestrators; and know how to negotiate the use of their resources. Thus, requiring extensive modifications and further development on existing orchestrators, as well as introducing considerable communication and negotiation overheads. Rather than taking this costly approach, we propose the introduction of a new entity above the distributed orchestrators, the hyperstrator, which is in charge of interacting with the underlying orchestrators and coordinating the lifecycle of NSs. In this way, not only each orchestrator becomes oblivious to the existence of other orchestrators and segments, but we can potentially integrate existing orchestrators present in real network deployments under the hyperstrator, through the development of simple, bespoke translation layers between existing northbound configuration interfaces (used by NPs to configure their networks) and the hyperstrator.

The combination of the hyperstrator and the distributed orchestrators acts as a single E2E network orchestrator, responsible for managing the entire E2E network infrastructure in the administrative domain of an NP. The hyperstrator itself is the interface for interacting with the E2E network, the central point for (*i*) instantiating customised NSs leveraging heterogeneous resources available across multiple network segments; (*ii*) monitoring existing communication services by gathering performance metrics from the distributed orchestrators; and if necessary, (*iii*) optimising the use of network resources according to the demands of communication services. Therefore, the hyperstrator serves as a network automation tool that not only facilitates network operations for NPs to deploy NSs, but also enables new business models, allowing NPs to offer NS as a Service (NSaaS) and automatically provision NSs to serve Service Providers (SPs), e.g., tenants and verticals [4]. Also, one hyperstrator could act as an SP to lease resources from hyperstrators responsible for other administrative domains, e.g., requesting RAN slices to provide coverage in different areas, or CN slices to offload computations during peak demand. Therefore, using a hyperstrator per administrative domain could facilitate and standardise resource sharing among NPs.

B. Translating and Delegating E2E Requirements

The SPs can request NSs to the hyperstrator, specifying them using NS descriptors, manifests containing high-level E2E requirements [13], e.g., mobile coverage on particular areas with connectivity to given data centres with certain capacities. Possible implementations of the descriptors include JSON and YAML, using YANG or TOSCA modelling languages [11]. In parallel to NFVO-C [10], the hyperstrator is responsible for translating these high-level E2E requirements into requirements for specific network segments, in the form of NSS descriptors, e.g., RAN and CN slice descriptors. However, in contrast to NFVO-C, these descriptors are tailored to the capabilities of each segment [11], e.g., containing the required throughput, latency and reliability, as well as coverage areas and points of exchange, respectively. NPs can specify or develop different translation mechanisms for their hyperstrators, such as the one detailed in [13]. The hyperstrator forwards these NSS descriptors to the respective underlying orchestrators, as shown in Fig. 2. Each orchestrator uses its own resource management directives to map local requirements into low-level network configuration to create suitable NSSs, deciding the appropriate allocation of resources and placement of functions to fulfil the local requirements [14].

The resulting NSs may comprise multiple NSSs, each of which can possess chains of different types of functions, e.g., VNFs, implementing network services and upper layers of the communication stack; and Virtual Radio Functions (VRFs), implementing RATs and lower layers of the communication stack [12]. Depending on the allocated resources and the functions placed, each NSS will achieve different performance. The hyperstrator can guarantee consistent QoS for NSs by requesting NSSs that deliver the required E2E throughput, while remaining within the E2E delay and reliability budgets of the NS. However, some NSSs may have resource and performance limitations, which the hyperstrator can try to compensate by making a trade-off in the E2E resource allocation and requesting NSSs with more resources, and that deliver higher performance, from other segments.

C. Ensuring Consistency Across Multiple Network Segments

The distributed nature of our hierarchical orchestration architecture poses new challenges for the design and implementation of this solution, especially for the hyperstrator, which must ensure consistency across multiple orchestrators and segments. The system must cope with issues that may arise during the instantiation of NSs in different segments, e.g., lack of the necessary resources, failures in the resource allocation or functional placement, and unexpected communication or hardware failures. We can circumvent such failures using a transactional communication protocol between the hyperstrator and the underlying orchestrators, where: the hyperstrator requests the execution of operations on multiple network segments, and according to the results of such requests, the hyperstrator either commits the operations, making the orchestrators carry out the commands, or rolls them back, reverting the instantiation of NSSs. This approach enables persistent, atomic operations over NSs, while ensuring consistency across multiple network segments. Furthermore, NSs should possess a unique identifier across all networks segments, e.g., Universally Unique Identifier (UUID), facilitating operations over NSs, access to their information, and the implementation of authorisation mechanisms for managing the NSs, e.g., Access Control Lists.

Our proposed architecture differs from other hierarchical orchestration approaches in the literature [7]–[10], as we do not centralise the decision over the E2E network management in a new entity, responsible for managing certain types of segments. In reality, the hyperstrator does not have any role on the resource allocation and function placement on particular segments. Instead, the hyperstrator (*i*) translates global service requirements for creating NSs into local requirements for specific segments; (*ii*) delegates local requirements to the respective specialised orchestrators, which are free to adopt the state-of-the-art or proprietary resource management solutions for creating customised NSSs; and most importantly, (*iii*) ensures cohesive performance across NSSs for guaranteeing consistent QoS for NSs. Hence, our proposed architecture leverages the capabilities of existing orchestrators and the domain expertise of their established communities to provide the most effective resource management in every segment.

IV. Experimental Proof of Concept

In this section, we provide an example deployment of NSs using the hyperstrator for coordinating separate specialised orchestrators. First, we describe a proof-of-concept implementation of our hierarchical orchestration scheme, used

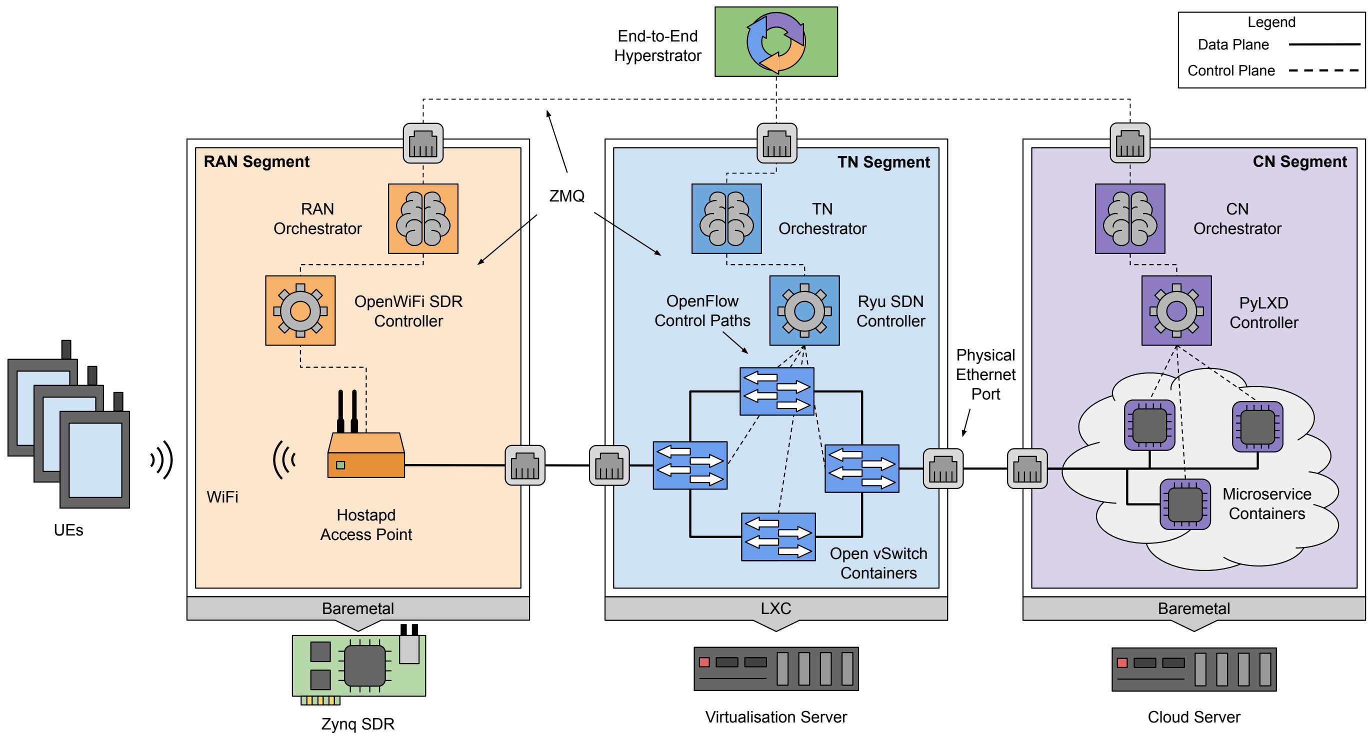

Fig. 3: The experimental setup we developed to validate our hierarchical orchestration architecture.

for managing an experimental E2E network infrastructure. Then, we assess the overhead introduced by our hierarchical orchestration architecture to provision NSs, and the impact of each NSS on the E2E performance of NSs.

A. Building the Experimental Setup

To verify the feasibility of our proposed architecture, we created an experimental E2E network infrastructure that resembles mobile networks, consisting of RAN, TN and CN segments managed by a hierarchy of orchestrators, as illustrated in Fig. 3. These network segments connect clients with their User Equipments (UEs) to desired microservices in the CN. The prototype hyperstrator serves as the central point for managing our E2E infrastructure and deploying NSs. It provides a Create, Read, Update, Delete (CRUD)-based interface, where SPs can instantiate, query, modify or remove NSs. The SPs must specify the requirements for NSs in the form of NS slice descriptors, JSON data structures containing high-level E2E requirements. In possession of such information, our prototype operates as follows.

- The hyperstrator translates E2E requirements into requirements specific for each network segment. In this case, considerations include coverage to UEs in the RAN, computing to host services in the CN, and paths between both in the TN.
- Then, the hyperstrator requests the instantiation of NSSs with tailored requirements to the underlying orchestrators. Each orchestrator is free to adopt their resource management directives, protocols and internal processes.
- Upon successful creation of the necessary NSSs, the NS becomes operational and the UEs can communicate with a new microservice. The hyperstrator returns the UUID of the NS to the SP, who can use it to manage the NS.

To simplify our initial prototype, we employed the ZMQ messaging library for both the northbound communication with SPs and the southbound interface towards the underlying orchestrators, as well as created homebrewed orchestrators for managing each network segment. These orchestrators are responsible for basic dimensioning of the required resources to meet local service requirements, and communicating with the hyperstrator and the respective underlying controllers. In the future, we plan to extend ONOS and OSM, adding new capabilities for supporting hierarchical orchestration. Moreover, we used Linux Containers (LXC) on servers at the Iris Testbed (http://iristestbed.eu/) for realising the majority of elements in our setup, reducing the physical size and overall complexity of our experimental infrastructure.

We leveraged software-defined hardware platforms as enablers for network slicing [9] and set up an SDR-based RAN segment, SDN-based TN segment, and LXC-based CN segment, shown in Fig. 3. The RAN segment is composed of a Zynq SDR running the OpenWiFi SDR controller [15], allowing us to create a Hostapd access point with tailored resource allocation per UE. We create RAN slices using OpenWiFi's slicing mechanisms to allocate non-overlapping portions of airtime for meeting local requirements. The TN segment is composed of four Open vSwitch containers forming an emulated ring topology in a virtualisation server. Two switches are also attached to Ethernet ports on the Zynq SDR and the cloud server, serving as points of exchange towards the RAN and CN segments, respectively. We create TN slices using Ryu SDN controller's features for creating overlay net-

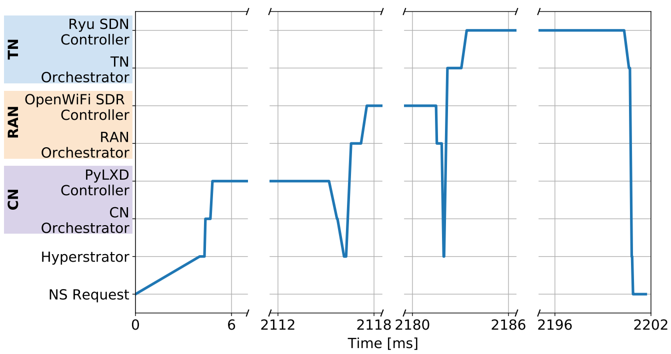

Fig. 4: Timing diagrams showing the interactions between the entities of our experimental setup required for deploying NSs.

	RAN Radio Airtime		TN Link Capacity		CN CPU Cycles	
	Throughput [Mbps]	Delay [ms]	Throughput [Mbps]	Delay [ms]	Throughput [Mbps]	Delay [ms]
100%	21.8	1.97	22.3	2.41	24.5	16.982
90%	17.9	2.10	22.2	1.86	**23.2**	**41.67**
80%	16.8	3.09	22.0	2.10	20.5	105.10
70%	14.1	4.35	22.7	1.75	18.6	147.59
60%	**11.5**	**6.48**	22.7	2.07	16.1	191.16
50%	8.47	8.07	21.5	2.18	13.2	219.32
40%	6.41	19.41	21.3	2.13	11.2	372.04
30%	5.44	35.24	22.2	2.14	8.40	466.51
20%	4.02	43.40	19.8	2.04	6.13	639.77
10%	2.18	61.37	18.7	2.01	3.03	875.99
1%	0.43	124.27	**9.88**	**1.98**	1.20	998.40

TABLE II: Example of how the allocation of distinct types of resources in different network segments affects the E2E performance of the NSs. We can use the hyperstrator to coordinate the different orchestrators to establish a dedicated 10 Mbps NS with latency of 50 ms using 60% of the radio resources, 1% of the transport resources, and 90% of the compute resources (shown in bold), while still leaving available resources for creating more NSs tailored to other services.

works with tailored queue configurations. The CN segment is composed of a cloud server running LXD, allowing us to host microservices on containers with customised specifications. We create CN slices using PyLXD controller's features for instantiating customised containers from different images.

To combine our NSSs, we attribute IP addresses to new RAN and CN slices, and use new TN slices to interconnect them. In the RAN, we programmatically configure a DHCP server running in the Zynq SDR for assigning valid IP addresses to new UEs, and forward their traffic to the data plane Ethernet port. In the CN, for each container hosting a new service, we create a virtual network interface attached to a bridge network with automatic DHCP, connected to the cloud server's data plane Ethernet port. Then, we use the IP addresses of the RAN and CN slices as arguments for creating TN slices, where we calculate and establish the shortest path between all the routes that support the required throughput between these IPs. Also, we utilise *iptables* for blocking communication between UEs, and between containers, ensuring traffic isolation within the same segment. For further information, the reader can refer to the hyperstrator's repository (https://github.com/orca-project/hoen).

B. Overhead for Provisioning Network Slices

In this analysis, we are interested in the interaction between the elements of our hierarchical orchestration scheme and their delay for provisioning NSs. The total provisioning delay consists on the time interval required for deploying new NSs to support communication services, i.e., between SPs sending NS requests, until the NSs are available, which includes: translating requirements, dimensioning NSSs and allocating resources in all network segments. Fig. 4 shows the results of our measurements for deploying customised NSs. The hyperstrator took 2.2 seconds to fulfil the request and instantiate a new NS. After receiving the request, the instantiation of RAN, TN and CN slices accounted for 2.99%, 0.84% and 95.85% of the provisioning delay, respectively. The hyperstrator functionality and the sequential communication with the distributed orchestrators accounted for roughly 2 ms, adding an overhead of 0.08%. We could further reduce the total provisioning delay by preallocating containers, reducing it to 90 ms and allowing us to instantiate up to 11 NSs per second. These results show that the hyperstrator can orchestrate heterogeneous network resources for provisioning NSs near real-time, and the hierarchical orchestration introduces a negligible overhead regarding the total provisioning delay.

C. Consistent QoS across Network Segments

In this analysis, we assess how the resource allocation in each network segment affects the E2E performance of NSs. Due to the independence between different segments and their resource management directives, the hyperstrator must coordinate the underlying orchestrators for deploying NSs with consistent QoS across multiple segments. Table II shows the E2E performance in terms of throughput and round-trip delay for an NS traversing through the CN, TN, and RAN segments towards a UE. For each segment, we varied the amount of allocated resources, i.e., radio airtime in the RAN, link capacity in the TN, and CPU cycles in the CN, while allocating maximum resources in the other segments, and measured the experienced E2E performance between a UE and a microservice container. These results illustrate how the resource allocation in each NSS significantly impacts the E2E performance of NSs, motivating the need for coordination between the distributed orchestrators in charge of the different network segments to ensure consistent QoS.

V. Conclusions and Open Challenges

In this paper, we presented a hierarchical orchestration scheme for E2E networks, a paradigm shift from traditional E2E network orchestration solutions that centralise the intelligence for managing the network infrastructure in a single monolithic entity. Our hierarchical orchestration architecture is both modular and extensible, capable of supporting new types of segments and resources unforeseen at design time. We have shown that the distributed nature of our orchestration architecture introduces negligible overhead for provisioning NSs in our particular setting, and confirmed the need of a hyperstrator for coordinating network segments and ensuring consistent QoS for NSs.

Based on the discussion in this paper, we conclude that our orchestration approach breaks down the E2E network management and network slicing challenges per network segment and leverages the capabilities of existing orchestrators and their communities to achieve an E2E fine-grained resource allocation. Naturally, there are many open challenges for the research and standardisation communities, beyond those addressed in this work, e.g., identifying requirements, classes, and relations of ontologies for hierarchical orchestration communication protocols; defining a systematic translation of global service requirements into requirements for each segment; modelling and evaluating distributed resource management schemes using carrier-grade orchestrators; and assessing the trade-offs for selecting different types of segments to provide radio coverage, perform computations and establish paths between other segments.

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support from the European Horizon 2020 Programme under the grant agreement No. 732174 (ORCA). We also acknowledge support from the Science Foundation Ireland under the grant No. 13/RC/2077.

REFERENCES

[1] 3rd Generation Partnership Project, "3GPP TR 22.891: Feasibility Study on New Services and Markets Technology Enablers," 3GPP, Tech. Rep., Sep. 2016.

[2] 5G PPP Architecture Working Group. (Accessed 2020, Sep.) View on 5G Architecture. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf

[3] J. F. Santos *et al.* (Accessed 2020, Sep.) Orchestrating Next-Generation Services Through End-to-End Network Slicing. White Paper. The ORCA Consortium. [Online]. Available: https://orca-project.eu/wp-content/uploads/sites/4/2018/10/orchestrating_e2e_network_slices_Final.pdf

[4] 3rd Generation Partnership Project, "3GPP TR 28.801: Study on Management and Orchestration of Network Slicing for Next Generation Network," 3GPP, Tech. Rep., May 2017.

[5] M. Howard, "The Evolution of SDN and NFV Orchestration," Infonetics Research, Tech. Rep., Sep. Accessed 2020. [Online]. Available: https://investor.juniper.net/files/doc_downloads/2015/May/Latest%20Resources/2000604-en_v001_h27j2q.pdf

[6] H. Flinck *et al.*, "Network Slicing Management and Orchestration," Internet Engineering Task Force, Tech. Rep., 2017.

[7] L. Peterson *et al.*, "Central Office Re-Architected as a Data Center," *Communications Magazine*, vol. 54, no. 10, pp. 96–101, 2016.

[8] R. Riggio *et al.*, "Programming Abstractions for Software-defined Wireless Networks," *Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[9] A. Rostami *et al.*, "Orchestration of RAN and Transport Networks for 5G: An SDN approach," *Communications Magazine*, vol. 55, no. 4, pp. 64–70, 2017.

[10] European Telecommunications Standards Institute, "ETSI GR NFV 003: Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV," ETSI, Tech. Rep., Jan. 2020.

[11] K. Katsalis *et al.*, "Multi-domain Orchestration for NFV: Challenges and Research Directions," in *International Conference on Ubiquitous Computing and Communications and International Symposium on Cyberspace and Security (IUCC-CSS)*. IEEE, 2016, pp. 189–195.

[12] A. Maeder *et al.*, "Towards a Flexible Functional Split for Cloud-RAN Networks," in *European Conference on Networks and Communications (EuCNC)*. IEEE, 2014, pp. 1–5.

[13] X. Zhou *et al.*, "Network Slicing as a Service: Enabling Enterprises' Own Software-defined Cellular Networks," *Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.

[14] N. Nikaein *et al.*, "Network Store: Exploring Slicing in Future 5G Networks," in *International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. ACM, 2015, pp. 8–13.

[15] J. Xianjun *et al.* (Accessed 2020, Sep.) Open-source IEEE802.11/Wi-Fi Baseband chip/FPGA Design. [Online]. Available: https://github.com/open-sdr/openwifi

BIOGRAPHIES

Joao F. Santos is pursuing a PhD in CONNECT, headquartered at TCD. His research interests include end-to-end network slicing and orchestration.

Wei Liu is a Post-Doctoral Researcher with Ghent University, she has experience regarding spectrum analysis, interference prevention technology recognition, and time-sensitive wireless networks on SDR platforms.

Xianjun Jiao is a Senior Researcher with imec on system design and implementation since 2016. Before imec, he worked 10 years in industry-leading companies.

Natal V. Neto is pursuing a PhD at Federal University of Uberlandia. His research interests include self-organising networks and autonomic computing.

Sofie Pollin is a professor with the Electrical Engineering Department, KU Leuven. Her research interest include wireless communications and networks.

Johann M. Marquez-Barja is a professor at University of Antwerp–imec, leading the wireless team at imec-IDLab-UAntwerp, and the connectivity of the Smartcity and SmartHighway Testbeds located in Antwerp.

Ingrid Moerman is a professor at Ghent University and programme manager of the Deterministic Networking track at imec, where she is responsible for end-to-end QoS-aware connectivity.

Luiz A. DaSilva is the Bradley Professor of Cybersecurity at Virginia Tech, where he serves as Executive Director of the Commonwealth Cyber Initiative. He is an IEEE Fellow for contributions to resource management in wireless networks.