A Multi-domain Experimentation Environment for 5G Media Verticals

Polychronis Valsamas, Ilias Sakellariou, Sophia Petridou, Lefteris Mamatas
University of Macedonia, Egnatia 156, 546 36, Thessaloniki Greece
{xvalsama, iliass, spetrido, emamatas}@uom.edu.gr

***Abstract*—5G networks are considered as a main enabler for new services for Media & Entertainment (M&E) verticals, allowing ultra-low delays, high numbers of mobile clients, resource-demanding operation, and large-scale UHD streaming. This evolution requires low-complexity and flexible experimentation of new service orchestration and management mechanisms over multi-domain 5G ecosystems.**

In this paper, we introduce 5G-CDN, an experimentation facility that focuses on the next-generation Content Delivery Networks (CDNs), a paradigm for hosting and launching M&E services. We build on top of existing novel test-bed federations, such as the FED4FIRE, to enable large-scale, multi-domain experimentation, involving: (i) *end-to-end network slicing* over multiple infrastructure providers utilizing heterogeneous hardware and virtualization resources; (ii) *dynamic resource discovery and allocation* residing at both federated open-access and local test-beds; and (iii) experimentation with *modular media service orchestration mechanisms*, e.g., on content caching and service elasticity. We provide proof-of-concept results demonstrating content provisioning over allocated slices and dynamic resource discovery involving European and USA test-beds, where heterogeneous physical and virtual resources co-exist.

***Index Terms*—Multi-domain Orchestation, 5G Networks, Media Verticals, Content Distribution Services, End-to-End Slicing**

I. Introduction

Technological advancements, such as 4G networking, and the abundance of powerful mobile devices have revolutionized the M&E services' market, leading to consumers expecting enhanced interactive experience and high-fidelity content everywhere. Important media services, including the next-generation content delivery, are expected to operate globally or at large-scales, be deployed rapidly, communicate with low delays, be resource-demanding and able to support large numbers of mobile users. Furthermore, they should be adaptable to changes in the business goals of the service provider, diverse service requirements, the network conditions and respond gracefully to fluctuations in service demands. This requires advanced *service orchestration mechanisms* (e.g., efficient load balancing, optimal content caching, service elasticity). An example is an interactive large-scale Ultra High Definition (UHD) streaming service for people around the globe attending parallel crowded events, e.g., new year celebrations.

5G technology is expected to have a catalytic impact on M&E service delivery. Network slicing is a central concept to the 5G success, since it declares a shift from existing monolithic cellular network architectures to the creation of virtual networks tailored to the performance requirements of each particular service. A 5G slice integrates core/mobile edge cloud and network resources in an isolated, guaranteed, in terms of performance, end-to-end (E2E) virtual network, with fast deployment, advanced network management and support for diverse service classes. Such slices should be E2E, involve heterogeneous cloud deployments in terms of physical and virtual resources, i.e., to utilize available edge-cloud options close to the users. For example, a CDN service may involve both edge and core clouds and deliver content through containers and OpenStack Virtual Machines (VMs), respectively. In such a setting, advanced *resource orchestration* capabilities, e.g., dynamic resource discovery, are necessary for deploying and operating multi-domain slices.

5G networking technologies are currently advancing to their trial phase targeting a commercial deployment in two years [1]. A main challenge is to demonstrate the opportunities to vertical sectors, such as M&E. In this context, there is a need for flexible, realistic and holistic experimentation of CDN services, involving both *resource and service orchestration* aspects. It should be noted that content delivery, a key service for M&E, is also of high relevance to applications in other vertical markets, including the e-health, educational and advertising sectors.

In this work, we propose 5G-CDN, a relevant experimentation facility that is based on the following technical enablers:

- A GUI and YAML [2] schema realizing a high-level definition of CDN experiments and modular extensibility of resource and content-delivery service orchestration algorithms through the NodeRED tool (http://nodered.org/). This allows the experimenters to focus on particular mechanisms and their realistic evaluation, without the time-consuming process of implementing the full infrastructure stack corresponding to the experiments.
- A novel architecture, appropriate abstractions and interfaces that: (i) bridge the gap between CDN-based services, important 5G features, and general-purpose test-bed federations through a holistic approach; (ii) handle and hide the complexity of heterogeneous physical and virtual resources; and (iii) perform dynamic resource allocation and discovery over both federated and local test-bed resources.
- The novel FED4FIRE facilities [3] offering: (i) a representation of the multi-domain infrastructure providers

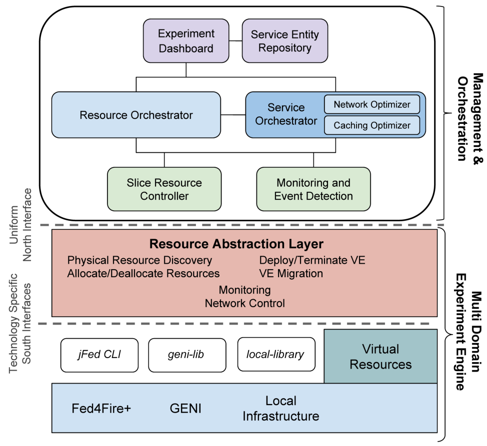


Fig. 1: The architecture of the proposed 5G-CDN platform

through the federated FED4FIRE test-beds providing hardware resources around the globe, including physical servers and 5G networking equipment; (ii) a basis for 5G network slicing through the FED4FIRE experimenters' slicing, in the sense of putting together physical resources from different test-beds, called slivers, to implement a particular service for experimentation; (iii) automations in the resource discovery and allocation through the FED4FIRE test-bed control tools (e.g., jFed CLI).

The paper includes proof-of-concept experiments based on a novel CDN scenario for content delivery through Unikernels [4]. Our results highlight the capabilities of the 5G-CDN to: (i) implement E2E slices over heterogeneous physical and virtual resources; (ii) dynamically discover resources for the CDN service deployment; (iii) define, in a flexible and modular manner, new experiments with alternative resource and service orchestration algorithms for CDNs.

The paper is organized as follows. Section II discusses the design and implementation details of the proposed platform, while Section III provides proof-of-concept results based on a novel CDN scenario. Section IV contrasts our solution against the related works and Section V concludes the paper.

II. Design and Implementation

The architecture adopted by the platform, depicted in Fig. 1, consists of two planes, the *Management and Orchestration* and the *Multi-Domain Experiment Engine*. The former provides all the necessary experimenter GUI, experiment definition and specification refinement, slice creation, control and management features, including advanced monitoring and results aggregation functionalities. The Engine's role is to provide uniform access to the federated test-bed resources, alleviating the need for low level, technology specific code provisioning on the experimenters' side. These two planes are detailed below.

A. Multi-Domain Experiment Engine

The bottom layer of the *Multi-Domain Experiment Engine* accommodates the physical resources located either on remote federated test-beds (i.e., currently FED4FIRE, but GENI [5] support is also under development), or our local infrastructure through custom relevant libraries. Accessing local resources through a federated experiment is important, since many important pilot 5G deployments are not part of large-scale federated test-beds (e.g., 5TONIC [6]).

The *Resource Abstraction Layer (RAL)* hides the resource and test-bed heterogeneity from the top architectural plane. *RAL* offers a technology agnostic uniform *North interface* to the upper *Management and Orchestration* plane, while translating incoming "north" operations to test-bed specific commands via its technology specific *South interfaces*, i.e., one for each diverse test-bed control approach. Moreover, *RAL* serves as a resource catalogue/provisioning layer for the experiments to be conducted, where resources are accessed by their corresponding test-bed control interface (e.g., jFed CLI or geni-lib). In practice, we maintain a local representation of the resources (in json format). This happens because both FED4FIRE and GENI represent their resources through Resource Specifications (called RSPECs [5]), but not in a uniform manner between the test-beds, e.g., the resources may have incomplete details or present different attributes.

The Virtual Entities (VEs), i.e, VMs and containers, are treated similarly to physical ones and are also accessible via the *RAL*. We support services operating over multiple Virtual Infrastructure Managers (VIMs), since it is common for edge and core clouds to use different virtualization technologies (e.g., OpenStack for core and Docker/Kubernetes for edge clouds). Finally, the *RAL* provides abstractions for diverse monitoring technologies for both slice resource and content delivery aspects, as well as SDN network control operations.

In short, the Engine provides a "universal" (i.e., physical & virtual), uniform (i.e., technology agnostic) and flexible (i.e., extendable to include new classes of resources) *North interface* to the high-level management and orchestration components. Due to these features, this plane enables experiments on dynamically discovered, heterogeneous resources.

B. Management and Orchestration

The *Management and Orchestration* plane follows a modular architecture that consists of the components depicted in Fig. 1 and detailed below.

The *Experiment Dashboard* is the platform's interface to the experimenter. It includes a GUI and along with the *Service Entity Repository* allow defining the service specifications and the details of each experiment (e.g., slice geographic constraints, KPIs, monitoring technology). In particular, the repository: (i) provides a set of visual components (for instance a Content Server VM), that can be added (drag-n-drop) to the *service graph* description, and (ii) assists the conversion of the *service graph* to a *slice graph* based on initial slice configuration details, that include but are not limited to the VIM requirements of the selected service entities, their resource requirements and network connectivity constraints. For example, a *service graph* represents the relations among service components, e.g., a load-balancer and a number of content servers are connected with "edges" that are annotated with requirements regarding service hosting components (e.g., VIM) or geographical and connectivity constraints. In the same example, the *slice graph* is the allocation of service components to slice parts, along with any network connectivity or physical/virtual resource requirements (e.g., number of physical machines, CPU / RAM demands), that will be populated during the slice creation phase. Graphs' representation is based on a custom YAML schema, inspired by TOSCA [7] and the ESpec (http://jfed.ilabt.imec.be/espec). The Dashboard also supports visualization of results and a bespoke visualization tool for large-scale CDN deployments.

The *Resource Orchestrator (RO)* component handles the population of the slice graph with physical resources. This slice building process involves the selection of appropriate resources among those available, requesting the allocation of resources and the necessary slice part stitching from the *Slice Resource Controller*. Operations involving slice resource elasticity, i.e., addition / removal of resources to existing slice parts and new slice part allocation along with the necessary stitching, also fall under the responsibility of the *RO* that decides the new slice configuration.

The *Service Orchestrator (SO)* offers service Virtual Entity (VE) deployment (i.e., of a VM or a container) on the allocated slice resources based on configurable placement algorithms, VE termination and operations involving service elasticity, e.g., VE migration or deployment as a response to service request events. The latter are detected via the corresponding component (*MED* - see below) in situations that require a graceful service elasticity operation. The two sub-components that handle this kind of events are the *Network Optimizer*, that deals with placement of VEs related to load balancing and traffic redirection in the slice, and the *Caching Optimizer*, that decides on the deployment of new VEs to respond to increased service requests. Finally, the *SO* handles the experiment scenario execution, i.e., starts/stops VEs emulating service clients' requests for video offered by a CDN service running on the slice. To perform these tasks the *SO* interacts directly with the *RAL*.

The *Slice Resource Controller (SRC)* discovers available resources with particular specifications through the *RAL* and handles the requests for physical resource allocation in the supported test-beds. In general, the *RO* takes the slice-level configuration decisions, while the *SRC* the slice-part-level (or test-bed-level) decisions. The *RO* and *SRC* are considered to be the main enablers for the E2E slice creation.

The *Monitor and Event Detection (MED)* module accesses monitoring data for physical and virtual resources via the *RAL* and detects traffic and other network events that one or both of the orchestrators should respond to. For instance, we carried out experiments on content popularity detection employing change point analysis [4], [8] triggering the *Caching Optimizer* component of the *SO* to deploy additional content-

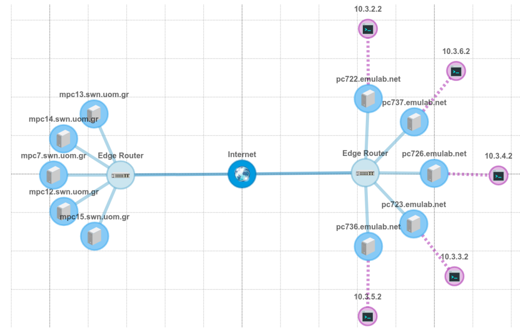


Fig. 2: Abstract view of the E2E slicing in our experiment

serving VEs. This component is also responsible for the results aggregation presented at the *Experiment Dashboard*.

All the above components besides the *RO* have been implemented through the NodeRED tool and according the microservices paradigm [9]. NodeRED is a browser-based flow editor wiring together independent software entities. Each entity is represented as a NodeRED node, i.e., a standalone Node.js component. For example, the addition of a new VE placement algorithm requires only a simple GUI task and minor changes in the placement decision mechanism through the embedded code editor.

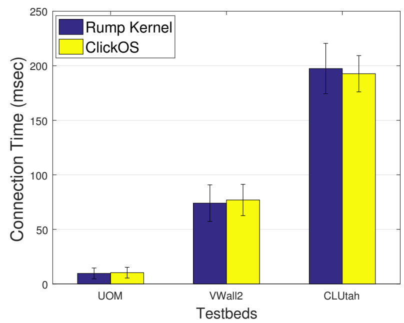
III. Proof-of-concept experimentation

Here, we demonstrate the functionality of the proposed 5G-CDN platform through a novel *CDN experimental scenario*, in the 5G multi-domain E2E slicing context. This scenario exploits the Unikernel technology, similarly to our work [4], to implement lightweight VMs, which are called Micro-Content Proxies (MCPs) and serve media data to the end-users.

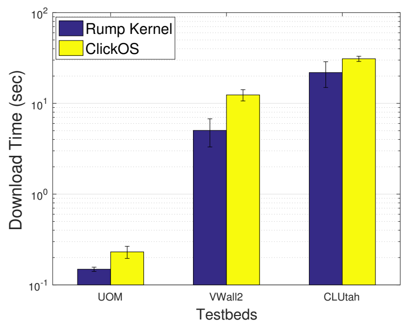
A. Experimentation Setup

Three different test-beds are utilized through the bottom layer of the *Multi Domain Experiment Engine*, namely: (i) the *Virtual Wall 2* (VWall2) test-bed which is part of the FED4FIRE federation and located in Europe; (ii) the *Cloudlab Utah* (CLUtah) test-bed, in Utah, USA; and, (iii) our local *UOM* test-bed in Greece, Europe. We build multi-domain E2E slices on top of these pieces of hardware considering two clusters of nodes, as illustrated in Fig. 2. The first cluster represents the east-end of a slice containing five physical nodes – always part of the *UOM* test-bed – which emulate the clients' behaviour initiating media service requests. The second cluster acts as the west-end of a slice consisting of six physical servers, five to host the MCPs and one to serve as an edge router. Allocating this west-end slice part in different test-beds, we obtain experimental results for geographically distributed slices, e.g., *UOM-UOM*, *UOM-VWall2* and *UOM-CLUtah* slices. To make the above infrastructure available for our experiments, the *SRC* performs dynamic resource discovery and exposes specifications, such as the memory, number of cores, disk storage and NIC information of hardware found.

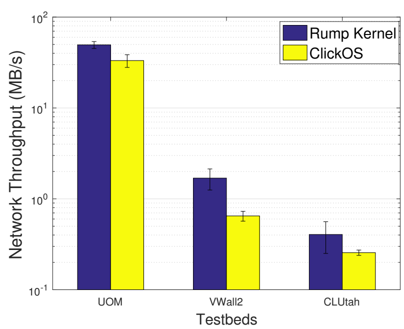
Apart from hardware, heterogeneity in our experimental setup is also reflected in the *RAL*, where different lightweight



(a) Clients' connection time

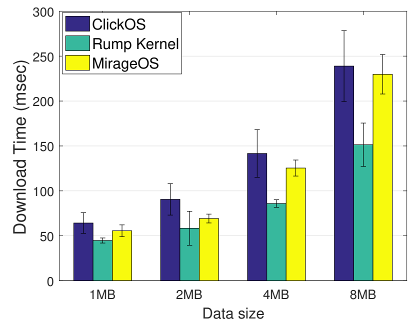


(b) Clients' download time

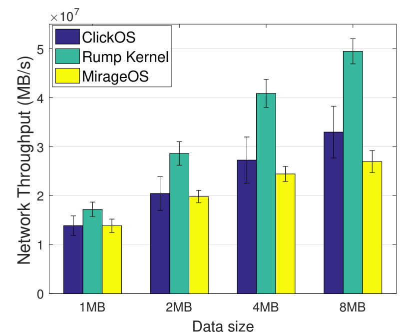


(c) Network throughput

Fig. 3: Different Unikernel technologies in E2E slices



(a) Clients' download time



(b) Network throughput

Fig. 4: The different unikernel technologies are evaluated in line with the content size (MB)

virtualization technologies are offered, namely ClickOS [10], Rump Kernel [11] and MirageOS [12]. Fig. 2 depicts an instant of such VMs placement in the MCPs' side (west-end of a slice) orchestrated by the *SO Caching Optimizer* mechanism. The latter could base its decisions on the feedback of a Change Point Detection (CPD) mechanism of the *Monitoring Component*, providing early tracking of content popularity changes [4], [8]. A DNS-based load-balance mechanism assigns the clients' requests to the deployed MCPs, in a round-robin fashion. Below, we detail our proof-of-concept results derived by such a setup.

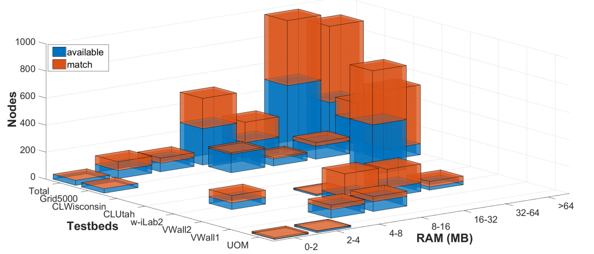
B. Experimental Results

1) Heterogeneity on E2E slicing: The first class of results show the capability of our platform to build E2E slices over multi-domain heterogeneous hardware and virtual resources.

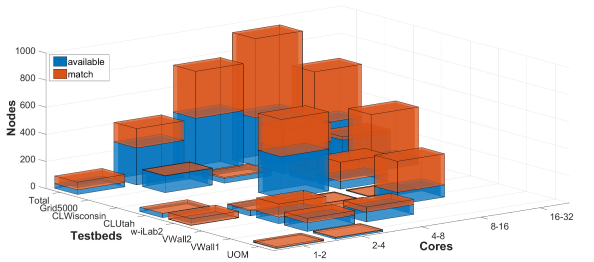
In Fig. 3 we evaluate the performance of MCPs hosting and launching data of 8 MB size, in terms of clients' *connection time* (i.e., the time elapsed between sending a media service request and receiving the fist response byte), *download time* (i.e., the time required for downloading to be completed) and *network throughput* (i.e., the ratio of the amount of data downloaded to the download time). We conducted the same experiment for each of the aforementioned slices, namely *UOM-UOM*, *UOM-VWall2* and *UOM-CLUtah* and we use either ClickOS or Rump Kernel as virtualization technology.

Fig. 3(a) and 3(b) validate the overall functionality provided by the 5G-CDN platform. It is expected that geographically remote slice-ends deteriorate users' experience. Indicatively, clients' connection time is roughly 10 $msec$ for requests being served locally, 75 $msec$ in the Greece-to-Europe slice and 200 $msec$ in the Greece-to-USA slice. The "geographical" distance is also reflected in the data download time which ranges from 0.1 sec to 32 sec. Obviously for fixed data size of 8 MB, lower download time entails higher network throughput, which is illustrated in Fig. 3(c). Regarding the Unikernels' technology, we observe deviation in the performance due to differential implementation approach for each unikernel. We notice that Rump Kernel outperforms ClickOS in download time and network throughput, while they exhibit similar performance in respect to the clients' connection time.

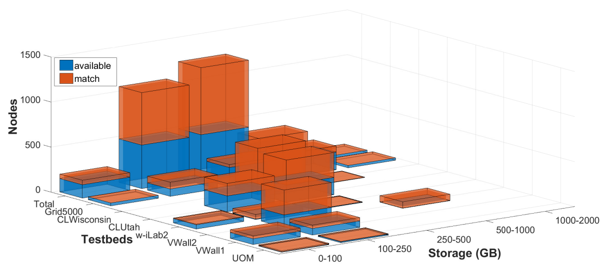
We further elaborate on VMs technology in Fig. 4. In this case, we test the performance of ClickOS, Rump Kernel and



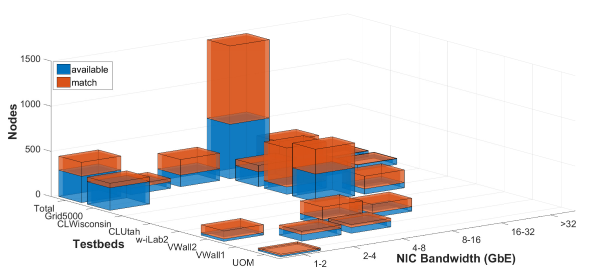
(a) Memory



(b) Number of cores



(c) Disk storage



(d) NIC specifications

Fig. 5: Resource discovery results in different testbeds

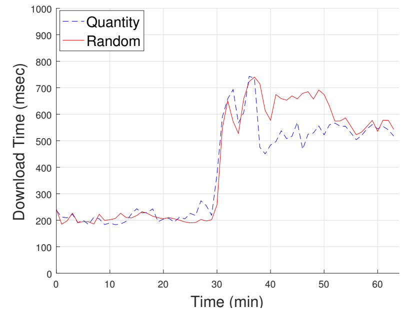


Fig. 6: The placement algorithms' performance

MirageOS technologies assuming that both MCPs and clients reside on the *UOM* test-bed and the MCPs serve data of $1, 2, 4$ and 8 MB. Our results show that the download time (Fig. 4(a)) and network throughput (Fig. 4(b)) are straightly associated with the data size independently of the Unikernel technology used, while Rump Kernel outperforms other technologies in respect to both metrics, especially for high data volumes. This outcome could lead to a VM placement strategy selecting the Rump Kernel technology, when data size is augmented. However, such behavior requires further investigation.

2) Resource discovery: The second class of results show the resource discovery facility offered by the proposed 5G-CDN platform. For example, when clients' interest for media content suddenly increases new VMs should be deployed to deliver the content; however the decision for their placement should be tailored to the available resources.

Fig. 5 presents the outcome of the *SRC* discovery process, for physical resources satisfying specific memory, CPU, disk storage and NIC bandwidth criteria, in different test-beds, which apart from the UOM, VWall2 and CLUtah include Grid5000, CLWisconsin, w-iLab2 and VWall1. Hardware that simply matches these criteria and resources that are actually available during the discovery process is depicted in different color. The data plotted in Fig. 5(a)-5(d) provides information necessary for *SO* VM placement decisions. Additionally, such information also assists in experiment setup, indicating test-beds that can act as core clouds, since they contain high-end resources, and low-end resource test-beds that could operate as edge-clouds.

3) Modular service orchestration: Our last outcome demonstrates in real time an example of a modular service orchestration mechanism available in our platform. More precisely, it is a preliminary result depicting the performance of two placement algorithms, i.e., *quantity* and *random*, in respect to the download time experienced by the end-users.

Once the *RO* informs the *SO* for a pool of available resources, a placement algorithm (part of the *Caching Optimizer*) should decide to deploy additional MCPs on the allocated resources. Obviously, a *random* choice cannot perform better compared to one that takes into account the number of existing VMs (*quantity*) already running on the candidate resources (Fig. 6). Our next steps include: i) more sophisticated algorithms optimizing either resources (e.g., memory, CPU) or KPIs derived by the slice *Monitoring Component*; ii) large-scale experiments to investigate the capacities of the proposed platform under stress condition and multiple infrastructure providers.

IV. Related Works

Here, we contrast our approach to the related 5G platforms: (i) utilizing the FED4FIRE or GENI federations; and (ii) supporting E2E slicing and experimentation for vertical services, especially media-related.

Relevant experimentation environments exploiting FED4FIRE or GENI capabilities are: (i) FUTEBOL [13] integrating wireless and optical domains over European and Brazilian test-beds; (ii) SoftFIRE (http://www.softfire.eu) an SDN/NFV test-bed supporting high-level service definition based on TOSCA [7] and resource discovery; and (iii) 5GinFIRE [14] a 5G test-bed targeting multiple vertical industries (one of the 5GinFIRE test-beds, the TNO 5G Media Vertical, focuses on the Media Vertical industry).

5G platforms with inherent 5G E2E slicing capabilities include: (i) the 5G-VINNI (http://5g-vinni.eu) targeting particular 5G KPIs and multiple verticals; (ii) the 5G-PAGODA [15] investigating NFV-based E2E slicing over two test-beds in Europe and Japan; and (iii) 5G-EVE (http://5g-eve.eu) which is an experimentation-oriented platform for multiple verticals, supporting a cross-facility 5G catalogue and multi-domain orchestration. Other proposals focus on RAN slicing aspects (e.g., [16]). The 5G-MEDIA [17] project and platform targets the Media Verticals through multiple use-cases, focuses on SDN/NFV aspects and supports a DevOps environment for media applications, hiding the complexity of service development and deployment over 5G infrastructures.

To the best of our knowledge, 5G-CDN is the first 5G platform tailored for CDN services, exploiting the novel FED4FIRE experimentation capabilities, while addressing multi-domain operation, modular service and resource orchestration, scalability and heterogeneity aspects. Our approach does not focus on RAN slicing, but it can incorporate relevant capabilities offered by existing FED4FIRE test-beds, such as the LTE slicing feature of the NITOS test-bed [18].

V. Conclusions

This paper presents the 5G-CDN platform, a novel experimentation facility for large-scale, multi-domain E2E slicing for media content delivery. It is built on top of the FED4FIRE to exploit its assets of large-scale experimentation with heterogeneous hardware, resource-facing slicing and novel test-bed control. It extends the FED4FIRE slicing concept to cover the service and virtualization aspects and enable relevant automation (e.g., dynamic resource discovery and service-aware E2E slice establishment). It also abstracts test-bed control approaches, multiple virtualization technologies (e.g., VIMs) and resource specifications (i.e., applying uniform representation of resources). The proposed platform refines a high-level service definition to lower-level slice specifications, while supporting modular CDN service orchestration. We are currently working on investigating alternative media service aspects, e.g., augmented reality scenarios, interactive services and streaming approaches. We also elaborate on alternative top plane architectures for other vertical services, on top of the same *Multi-domain Experiment Engine*.

Acknowledgments

This work has received funding from the EU's Horizon 2020 research and innovation programme through the 4th open call scheme of the FED4FIRE+ (grant agr. nº 732638) and the Horizon 2020 NECOS (grant agr. nº 777067) project. Special thanks to Brecht Vermeulen and Wim Van de Meerssche from IMEC for their support on the FED4FIRE facilities.

References

[1] NGMN Alliance, "5g white paper," Feb. 2015, pp. 1–125.

[2] V. Sinha, F. Doucet, C. Siska *et al.*, "Yaml: a tool for hardware design visualization and capture," in *Proc. of 13th Int. Symposium on System Synthesis (ISSS'00)*, Sep. 2000, pp. 9–14.

[3] T. Wauters, B. Vermeulen, W. Vandenberghe *et al.*, "Federation of internet experimentation facilities: architecture and implementation," in *European Conf. on Networks and Commun.*, Jun. 2014, pp. 1–5.

[4] P. Valsamas, S. Skaperas, and L. Mamatas, "Elastic content distribution based on unikernels and change-point analysis," in *Proc. 24th Eur. Wireless Conf. (EW)*. VDE, May 2018, pp. 1–7.

[5] M. Berman, J. S. Chase, L. Landweber *et al.*, "Geni: A federated testbed for innovative network experiments," *Computer Networks, Elsevier*, vol. 61, pp. 5–23, Mar. 2014.

[6] B. Nogales, I. Vidal, D. R. Lopez *et al.*, "Design and deployment of an open management and orchestration platform for multi-site nfv experimentation," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 20–27, 2019.

[7] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "Tosca: portable automated deployment and management of cloud applications," in *Advanced Web Services*. Springer, 2014, pp. 527–549.

[8] S. Skaperas, L. Mamatas, and A. Chorti, "Early video content popularity detection with change point analysis," in *Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[9] P. Di Francesco, P. Lago, and I. Malavolta, "Migrating towards microservice architectures: An industrial survey," in *IEEE Int. Conf. on Software Architecture (ICSA)*, 2018, p. Apr.

[10] J. Martins, M. Ahmed, C. Raiciu *et al.*, "Clickos and the art of network function virtualization," in *Proc. of the 11th USENIX Conf. on Networked Systems Design and Implementation (NSDI)*, Apr. 2014, pp. 459–473.

[11] J. Cormack, "The rump kernel: A tool for driver development and a toolkit for applications," in *Proc. of the Asian BSD conf.(AsianBSDCon)*.

[12] A. Madhavapeddy, R. Mortier, R. Sohan *et al.*, "Turning down the lamp: Software specialisation for the cloud," in *Proc. 2nd USENIX Conf. HotCloud*, 2010.

[13] P. Marques, C. Silva, V. Frascolla *et al.*, "Experiments overview of the eu-brazil futebol project," in *Proc. of the 26th European Conf. on Networks and Commun.*, Jun. 2017, pp. 1–2.

[14] A. Gavras, S. Denazis, C. Tranoris *et al.*, "Requirements and design of 5g experimental environments for vertical industry innovations," in *Global Wireless Summit (GWS)*. IEEE, Oct. 2017, pp. 165–169.

[15] I. Afolabi, A. Ksentini, M. Bagaa *et al.*, "Towards 5g network slicing over multiple-domains," *IEICE Transactions on Commun.*, vol. 100, no. 11, Nov. 2017.

[16] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture," in *ACM Proc. of the 23rd Annual Int. Conf. on Mobile Computing and Networking*, Oct. 2017, pp. 127–140.

[17] S. Rizou, P. Athanasoulis, P. Andriani *et al.*, "A service platform architecture enabling programmable edge-to-cloud virtualization for the 5g media industry," in *IEEE Int. Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2018.

[18] N. Makris, C. Zarafetas, S. Kechagias *et al.*, "Enabling open access to lte network components; the nitos testbed paradigm," in *Proc. of the IEEE conf. on Network Softwarization (NetSoft)*, Apr. 2015, pp. 1–6.