MESON: Optimized Cross-Slice Communication for Edge Computing

George Papathanail, Angelos Pentelas, Ioakeim Fotoglou, Panagiotis Papadimitriou
University of Macedonia, Greece
{*papathanail, apentelas, fotoglou, papadimitriou*}@*uom.edu.gr*

Konstantinos V. Katsaros, Vasileios Theodorou, Sergios Soursos
Intracom Telecom, Greece
{*konkat, theovas, souse*}@*intracom-telecom.com*

Dimitrios Spatharakis, Ioannis Dimolitsas, Marios Avgeris, Dimitrios Dechouniotis, Symeon Papavassiliou
National Technical University of Athens, Greece
{*dspatharakis, jdimol, mavgeris,ddechou*}@*netmode.ntua.gr, papavas*@*mail.ntua.gr*

***Abstract*—Network slicing is set out to address crucial needs of 5G, including support for multi-service provisioning. Focusing on resource and performance isolation, as well as security concerns associated with multi-tenancy, existing management and orchestration (MANO) frameworks typically offer network slices in the form of isolated bundles of computing, storage, and network resources, across the network infrastructure, including the edge. However, network slice isolation in its prevailing form raises significant concerns related to performance and resource utilization, hindering potential Business-to-Business (B2B) synergies.**

In this paper, we discuss a novel aspect of network slicing, *i.e.,* optimized cross-slice communication (CSC). We argue that multi-tenancy and service co-location open up unique opportunities for B2B interactions, inter-service communications and service composition, especially in the case of edge computing and location-based services. In this context, we present *optiMized Edge Slice OrchestratioN* (MESON), a MANO-based architecture for optimized CSC in edge clouds. We discuss the main architecture components and descriptors, as well as the steps required (i) for the matching between CSC offerings from providers and expressed CSC interests from potential consumers, and (ii) the establishment of optimized CSC upon matching. We further present experimental results from initial feasibility tests with CSC.

I. Introduction

Over the last years, there has been an increasing interest in network slicing, with various application domains, such as cellular networks and edge computing. Network slicing, enabled by Network Function Virtualization (NFV) [1], [2], [3] and Software-Defined Networking (SDN) [4], essentially allows network providers to lease bundles of computing, storage, and network resources to Service Providers, active within commercial domains often termed as verticals. This opens up a unique opportunity for services to get deeply integrated into the (edge of the) network infrastructure, realizing innovative applications with stringent performance requirements in various domains of social and economic activity, *e.g.,* automotive, media and entertainment, e-health, industry to name a few.

Taking a step further, and in an analogy to the well-established cloud realm, we expect these technological advances to facilitate the emergence of a broader service ecosystem, allowing cross-service interactions. Such interactions may significantly vary, from existing operations, such as single sign-on systems [5], to currently emerging Function-as-a-Service (FaaS) deployments [6], to future service interactions in the context of location-based (edge) services, *e.g.,* touristic guide applications integrating advertisement or social networking information [7].

In this context, we present a new management and orchestration (MANO) based architecture (namely MESON) for optimized and secure cross-slice communication within edge clouds, fostering cross-slice/tenant interactions for next-generation services. MESON provides the necessary means for the establishment of B2B and cross-service interactions, exploiting opportunities for service co-location. More precisely, MESON encompasses orchestration primitives for CSC service discovery, slice placement and CSC establishment, subject to Operations/Business Support System (OSS/BSS)-level procedures expressing the intent of service providers to establish a synergy. To this end, we define MEC-based descriptors for the expression of CSC offered services and intents, and we also explain the interactions between the MESON architecture components for CSC establishment. In addition, we study the feasibility of CSC setup, relying on existing NFV MANO platforms, such as OpenStack [8] and OpenSource MANO (OSM) [9].

The remainder of the paper is organized as follows. In Section II, we elaborate on the concept and the potential gains from CSC. Section III provides an overview of MANO, whereas Section IV presents the MESON orchestration architecture and initial feasibility tests. In Section V, we briefly discuss use cases for CSC. Finally, Section VI highlights our conclusions.

II. Cross-Slice Communication

Network slice provisioning is commonly mandated by isolation, which, in a strict form, hinders communication optimizations between network slices and the corresponding service components, at the event of service co-location. For instance, a Video-on-Demand (VoD) service may need to combine the playback of a video file with an overlaid personalized online advert, offered by a third-party service provider.

Facilitating cross-slice communication, by confining traffic within datacenter (DC) boundaries, becomes very important when the compute/storage infrastructure is located at the edge of the network, *e.g.,* close to a base station. In such cases, exiting network slice boundaries typically requires the traversal of the entire mobile network infrastructure so as to reach the mobile network gateway (which is not the case when service components reside closer to the gateway). This is deemed more crucial, when considering the focus of edge computing on the support of low-latency applications. Besides latency savings, optimized CSC is further expected to reduce the traffic volume in the backhaul/transport network.

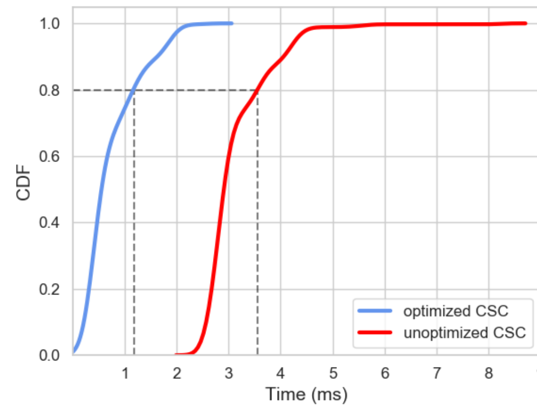

Fig. 1: CDF of measured delay with optimized and unoptimized CSC.

In the following, we provide a schematic high-level illustration of the potential latency gains from optimized CSC. To this end, we deploy two virtual machines (VMs) in an edge cloud infrastructure (Stackpath [10]). These VMs correspond to service components of two different (but co-located) network slices. In addition, we deploy a third VM in the same region, using Amazon EC2. This VM essentially acts as a (mobile) core gateway, which routes traffic between the two other VMs, in the case of unoptimized CSC (where traffic is forced to leave the edge cloud). Our goal is to compare the latency between optimized (direct communication between the two VMs) and unoptimized CSC (communication through the core gateway).

According to Fig. 1, although in both cases latency appears to be low, for delay-sensitive applications the advantage of optimized CSC is notable. We specifically observe that for 80% of the delay measurements, optimized CSC yields a latency saving of approximately up to 70%. We further note that unoptimized CSC tends to skew the tail of the latency distribution, as opposed to optimized CSC, at which the latency is bounded at approximately 3*ms*. We hereby stress on the fact that for latency-critical applications (which is a typical case for edge computing), low latency is not only important on an average basis but also on a max basis. In this respect, unoptimized CSC cannot comply with such stringent delay budgets (*e.g.,* compromising safety in automotive applications).

III. MANO Primer

Since the proposed CSC orchestration framework is based on ETSI NFV MANO, we provide background information on MANO and the associated descriptors. MANO provides management and orchestration functionality across two layers: (i) *the Virtualized Infrastructure Manager (VIM)*, which, as its name implies, is responsible for the unified management of the physical infrastructure across all resource types (*i.e.,* compute, storage, network), and (ii) the *Orchestration* layer, which encompasses modules for service and resource orchestration, such as life-cycle and VNF state management.

We also briefly discuss the main constructs of MANO's information model. MANO uses the Virtual Network Function (VNF) as the lowest structural element that manages and orchestrates, specified by its descriptor, *i.e.*, `vnfd`. In essence, the `vnfd` is a configuration template which includes attributes, such as the virtualized resource requirements of the respective VNF. MANO further defines network services, as a composition of VNFs. To this end, MANO utilizes the Network Service Descriptor (`nsd`), which is also a configuration template. As such, a `nsd` is associated with `vnfds` along with the virtual links that connect them. In essence, the `nsd` specifies the service graph. Finally, the Network Slice Template (`nst`) is the top-level construct that describes a network slice, which, in turn, encompasses a set of inter-connected network services. Amongst various attributes, the `nst` can be associated with a particular service or a traffic class, in the context of 5G, such as enhanced Mobile Broadband (eMBB), ultra-reliable low latency communications (URLLC), or massive IoT (MIoT).

IV. MESON Architecture

In this section, we discuss the proposed MANO-based architecture, namely MESON, which facilitates the establishment of optimized communication between network slices, co-located in the same edge cloud infrastructure. MESON is currently applicable to a single edge cloud provider with multiple Points-of-Presence (PoPs).

A. Business Roles

The proposed MESON architecture has been specified with the following business roles in mind:

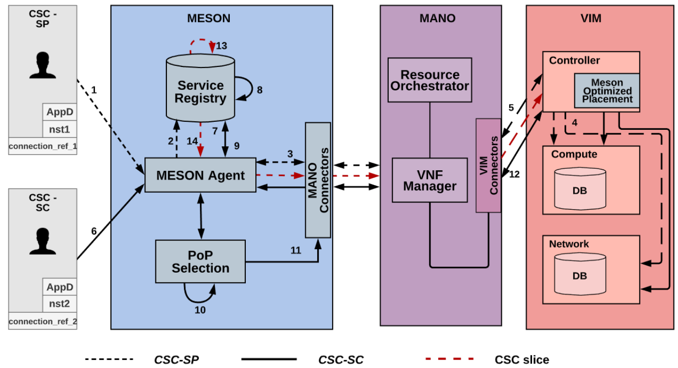

Fig. 2: MESON architecture.

Infrastructure Provider, who owns the physical cloud infrastructure and offers resources under lease basis for slice deployment.

CSC-enabled Service Provider (CSC-SP), who is the tenant of a network slice instantiated by the Infrastructure Provider. The CSC-SP also offers CSC-enabled services in selected PoPs to other slice tenants, which we call CSC-enabled Service Consumers.

CSC-enabled Service Consumer (CSC-SC), who is a slice tenant that wishes to establish communication with another slice, co-located in the same edge cloud, by the means of optimized CSC.

We note that both CSC-SP and CSC-SC are slice tenants, *i.e.*, clients of an Infrastructure Provider.

B. Architecture Overview

Within the context of MESON, the realization of CSC raises the following key functional requirements: (i) the advertisement of CSC offerings by a *CSC-SP*, (ii) the expression of interest from a *CSC-SC* for the establishment of connectivity with a co-located *CSC-SP*, (iii) the binding of CSC interests with offerings for CSC service consumption, (iv) communication path setup for CSC establishment between the slices of the CSC-SP and CSC-SC. The latter (*i.e.,* path setup) may be jointly optimized with the placement of the CSC-SC slice.

In order to meet these requirements, we have specified a three-layer MANO-based architecture, as depicted in Fig. 2. The two lower layers essentially encompass the management and service/resource orchestration functionality of MANO. In particular, at the bottom, MESON relies on a VIM, such as OpenStack, for the unified management of the

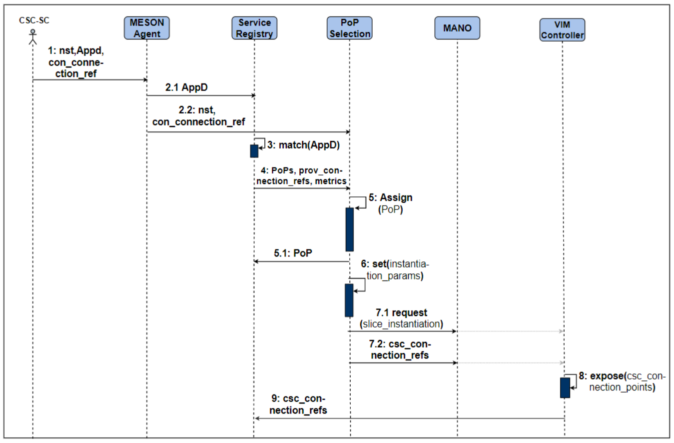

Fig. 3: Sequence diagram for CSC establishment.

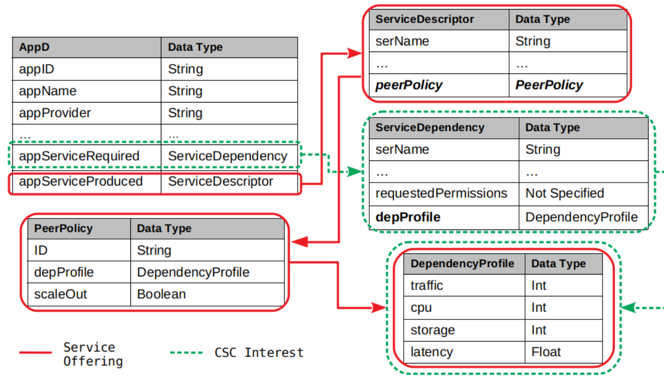

Fig. 4: Descriptors for CSC interests and service offerings.

physical infrastructure, whereas MANO's orchestration modules comprise the middle layer. The main functionality of MESON resides on the top layer, including the following components:

Service Registry, which handles the binding of CSC interests with CSC offerings, subject to an exposed policy, which we discuss in further detail in Section IV-C. To this end, the Registry maintains a list of CSC service instances offered by CSC-SPs. The interests expressed by CSC-SCs are also directed to the Registry.

PoP Selection, which identifies the most suitable PoP for the deployment of a slice requested by a CSC-SC. This module essentially generates a PoP ranking, based on service co-location and other KPIs advertised from individual PoPs.

MESON Agent, which comprises the client-facing interface for MESON. Both CSC service advertisements and interests are received by MESON via this agent.

We note that certain operations, such as slice placement optimization and communication path setup for optimized CSC, require additional functionality at the VIM layer, which we do not discuss in this paper, due to lack of space. Instead, we primarily focus on the MESON layer functionality and the interactions with the main MANO and VIM components for CSC establishment between CSC-SPs and CSC-SCs.

C. Service Discovery

Service discovery comprises a prerequisite for CSC establishment, as it performs the binding of CSC interests with service offerings from CSC-SPs. This process is handled by the *Service Registry*. In order to convey the required information from CSC-SPs/CSC-SCs (via OSS/BSS) to the *Service Registry*, we extend the existing descriptors provided by ETSI MEC specifications [11]. As shown in Fig. 4, MESON employs the MEC Application Descriptors (`AppD`) for the description of service attributes. The registration of a CSC service is established through the `appServiceProduced` field, which is of type `ServiceDescriptor`, whereas a CSC interest is expressed through the `appServiceRequired` field, which is of type `ServiceDependency`.

We further extend the `ServiceDescriptor` type with a `PeerPolicy` field used to describe the policy of CSC-SP, concerning service provisioning for CSC. A peering policy contains a list of policy-related properties, such as the highest acceptable computing and network resource values (described as a new `DependencyProfile` type) and the scale-out scheme (if any) supported by the CSC-SP in response to increased cross-slice traffic load. On the request side, we extend the `ServiceDependency` type with a `DependencyProfile` field describing the expected load of the intended CSC, as well as the desired latency of the communication path between the corresponding CSC counterparts.

Concerning the binding process, the *Service Registry* employs a mechanism that associates CSC interests with available services in order to identify the various CSC establishment options. This association is based upon the `DependencyProfile` fields, which have been described in the respective `AppDs`. The binding relies on a filtering feature that discards any slice instances that do not comply with the CSC criteria.

D. CSC Establishment

We hereby discuss the steps taken by MESON for CSC establishment. We distinguish between three phases, each one associated with the facilitation of a unique functional requirement. For brevity, in our illustrations (Figs. 3, 5) we refer to the network slice template (`nst`) that encompasses all descriptors that comprise the slice (*i.e.*, `vnfd`, `nsd`). The `connection_ref` attribute is used as a reference to the slice connection points, exposed for CSC establishment.

During the initial phase, the *CSC-SP* conveys the `nst`, `AppD` and `connection_ref` to the *MESON Agent* ***(step 1)***, which, in turn, forwards the `AppD` and `connection_ref` to the *Service Registry* ***(step 2)***. Since the instantiation of *CSC-SP's* slice is not dependant on other running slices (*i.e.,* the deployment of the two communicating slices is handled by MESON in an asynchronous manner), the *MESON Agent* directs the respective `nst` to the MANO layer for the slice deployment ***(step 3)***. Subsequently ***(step 4)***, the slice is assigned to a PoP, where it is deployed by the respective VIM. Furthermore, the slice is associated with a connection point, which is exposed for the purpose of optimized communication with another slice. This information is inserted into the *Service Registry* ***(step 5)***.

The second phase initiates with the submission of the CSC interest from the *CSC-SC*. This is expressed via the `AppD`, `nst` and `connection_ref`, all of which are submitted to the *MESON Agent* ***(step 6)***. Since MESON seeks the co-location of communicating slices, the placement of the *CSC-SC's* slice will depend on the deployment (*i.e.,* PoP) of the *CSC-SP's* slice. As such, a lookup is performed in the *Service Registry* ***(step 7)***, based on the *CSC-SC's* `AppD`. The *Service Registry* is responsible for matching CSC interests with registered CSC service instances, while adhering to *CSC-SP* and *CSC-SC* peering policy constraints (as expressed in their respective `AppDs`). This process is illustrated in ***step 8***. The outcome of this matching, *i.e.*, tuples of PoPs with references to their corresponding connection points, is conveyed to the *PoP Selection* module ***(step 9)***. The latter ranks the different CSC options with respect to KPIs advertised from the PoPs (*e.g.*, service response time, availability, cost, and scalability). These criteria can be further combined with QoE metrics. Eventually, MESON applies a multi-criteria decision making method for computing the final ranking of the PoP candidates ***(step 10)*** [12]. This concludes the configuration of the instantiation parameters of the *CSC-SC's* slice, thus enabling its deployment ***(step 11)***. In the final step ***(step 12)*** of this phase, the exposed connection points (at which the CSC slice will be attached) are stored in the *Service Registry*. Fig. 3 illustrates the sequence of messages exchanged between the MESON architecture components, as well as the methods invoked by each module.

The third and final phase consists of the instantiation of an intermediate slice for CSC. More specifically, to achieve secure, efficient and controlled communication among different slices, we introduce the concept of a dedicated CSC slice, *i.e.,* an intermediate network entity (operated by the Infrastructure Provider) purposed for cross-slice communication and policy control (Fig. 5). A CSC slice can incorporate additional functionalities, such as deep packet inspection (DPI), monitoring, and/or billing. The configuration of the CSC slice requires former knowledge of the connection points of the two slices that seek to communicate, as well as the PoP that hosts them. Such information has been previously stored in the *Service Registry* during the instantiation of each slice. In MESON, we consider several types of CSC slice templates stored in the *Service Registry*. This emphasizes the uniqueness of each CSC, as well as the need for addressing specific requirements that might occur according to the respective peering policies. In this respect, ***step 13*** represents the configuration of the intermediate slice according to the above parameters. Finally, in ***step 14*** the generated `nst` is conveyed to the lower layers.

E. Experimental Evaluation

We hereby conduct a preliminary evaluation of CSC setup, as envisaged by MESON. In particular, we measure the instantiation time of a CSC slice on top of our VIM (*i.e.*, OpenStack Queens), which manages a range of compute nodes deployed on three Dell PowerEdge R440 servers, each one with 8 CPU cores at 2.1 GHz and 64 GB of RAM.

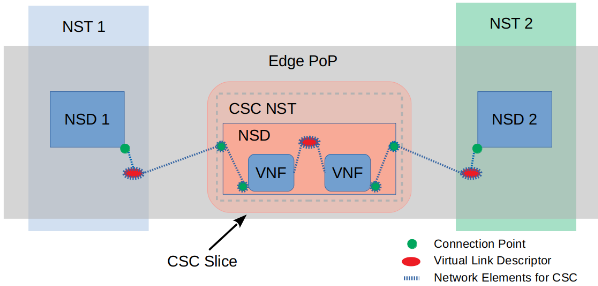

Fig. 5: Intermediate CSC slice attached to connection points of communicating slices.

Our main goal is to measure the time required for the instantiation of the intermediate CSC slice with a diverse number of VNFs, which may be used for monitoring, security or billing purposes, as explained in Section IV-D. In our measurements, the CSC-slice instantiation time is decomposed into *VNF creation time* and *connection setup time*. The former corresponds to the total time spent by Openstack in order to copy and create an instance of the corresponding image for the deployment of each VNF. *Connection setup time* is associated with the delay incurred during the CSC slice connection setup. This consists of the bridging of the CSC slice with the slices of CSC-SP and CSC-SC, as well as the VNF chaining (which applies to CSC slices with more than one VNF).

Fig. 6 illustrates that *VNF creation* is the dominant factor in terms of CSC-slice instantiation time, as opposed to the *connection setup time*, which comprises only a small fraction of the instantiation time. We further observe that all measured delays increase linearly with the number of VNFs. The increase of *connection setup time* is mainly attributed to VNF chaining, whose configuration incurs more delay with additional VNFs. In contrast, the time required for CSC-slice bridging remains constant, due to the fixed number of connections points. The relatively high VNF creation delay can be alleviated with VNF sharing among different CSC slices. This would obviate the need for the deployment of separate VNF instances for each required functionality within CSC slices. We plan to investigate the potential gains of VNF sharing in future work.

V. Use Cases

In this section, we discuss use cases for CSC.

Video Streaming. Video streaming occupies the lion's share of today's total Internet traffic. According to the latest Cisco Visual Networking Index (VNI) Complete Forecast, video streaming represents the 67% of all internet traffic, and it is projected to increase up to 80 by 2021. As expected, this puts not only the content providers but the network operators, as well, under severe stress. To alleviate this load, popular techniques such as Content Delivery Networks (CDN) have been utilized in the current network infrastructure. However, 5G brings even further possibilities, by allowing network services to be offered as VNFs deployed on micro-datacenters (μDC) across the (edge of the) network infrastructure. Taking advantage of the new 5G features, network operators can deploy their own (virtual) CDN services (vCDN) much closer to the end-user. In such an ecosystem, one can imagine multiple Virtual Network Operators (VNOs) offering CDN services which are co-located in the same μDC. Hence, VNOs can benefit from service-based interactions, offering and consuming video streaming services (storage, compression, transcoding, etc.) to/from other VNOs. In this as-a-service model, streaming services can be exchanged via direct communication of corresponding co-located VNFs, whereas VNOs can benefit from additional revenue streams and load balancing.

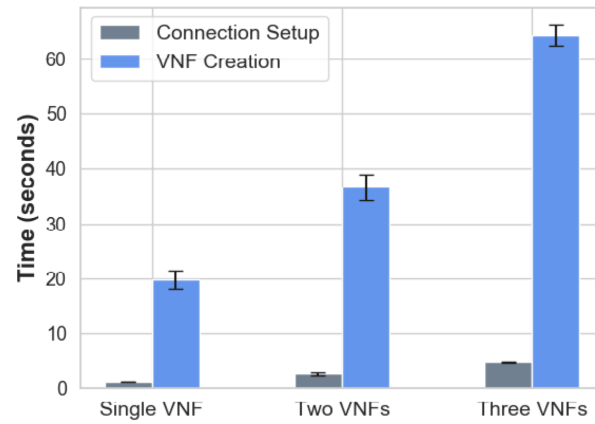

Fig. 6: CSC-slice instantiation time with a diverse number of VNFs.

Industry 4.0. One of the prerequisites and at the same time open challenges for Industry 4.0 is the physical interaction between robots in the factory floor in order to accomplish a specific task in a coordinated and secure way. Robotic processes (*e.g.*, localization [13]) rely on various sensors and complex algorithms that require plenty of computing resources. Despite the recent boost in the computing capabilities of robotic systems, local execution still remains a time-consuming process. For this reason, current trends in network service delivery dictate that robotic applications are being developed and treated, as VNFs, making their deployment to the cloud feasible. In such a setting, autonomous robotic clusters, deployed by different vendors in the same factory floor, can communicate through their corresponding slices in order to coordinate and perform their tasks more efficiently (*e.g.,* inventory loading/unloading).

VI. Conclusions

In this paper, we presented the architecture design of a new MANO framework that aims to enable optimized CSC and, thereby, open up new business opportunities for inter-slice interactions in edge cloud infrastructures. The potential gains from CSC are further reflected in two use cases discussed in the paper. As future work, we plan to perform an extensive feasibility study of CSC using a proof-of-concept implementation of MESON.

References

[1] NFV White Paper, "Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action. Issue 1," Oct. 2012.

[2] M. Kourtis *et al.*, "T-nova: An open-source mano stack for nfv infrastructures," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 586–602, Sept 2017.

[3] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[4] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[5] A. Pashalidis and C. J. Mitchell, "A taxonomy of single sign-on systems," in *Information Security and Privacy, 8th Australasian Conference, ACISP 2003*. Springer-Verlag, 2003, pp. 249–264.

[6] M. Villamizar *et al.*, "Infrastructure cost comparison of running web applications in the cloud using aws lambda and monolithic and microservice architectures," in *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. Springer-Verlag, 2016.

[7] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM conference on Recommender systems (RecSys '13)*. Springer-Verlag, 2013, pp. 93–100.

[8] "Openstack," http://www.openstack.org.

[9] "OSM," http://osm.etsi.org.

[10] "Stackpath - secure edge computing," http://stackpath.com, accessed: 2019-12-01.

[11] ETSI, "Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management," pp. 19–28, 2017. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/mec/001_099/01002/01.01.01_60/gs_mec01002v010101p.pdf

[12] Dechouniotis *et al.*, "Fuzzy multi-criteria based trust management in heterogeneous federated future internet testbeds," *Future Internet*, vol. 10, no. 7, p. 58, 2018.

[13] M. Avgeris *et al.*, "Single Vision-Based Self-Localization for Autonomous Robotic Agents," in *7th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2019, pp. 123–129.

George Papathanail is a PhD candidate at the department of Applied Informatics in the University of Macedonia, Greece. His research interests include network slicing, NFV and cloud computing.

Angelos Pentelas is a PhD candidate at the department of Applied Informatics in the University of Macedonia, Greece. His research interests include discrete optimization, NFV and network management.

Ioakeim Fotoglou is a PhD candidate at the department of Applied Informatics in the University of Macedonia, Greece. His research interests include (next-generation) networking, network & cloud Security, and 5G architectures.

Panagiotis Papadimitriou is an Assistant Professor at the department of Applied Informatics in the University of Macedonia, Greece. His research activities include (next-generation) Internet architectures, network processing, SDN, datacenter networking, and edge computing. Panagiotis is a Senior Member of IEEE.

Konstantinos Katsaros is a Senior R&D Engineer at Intracom S.A. Telecom Solutions, Greece, working on edge computing and NFV/MANO for 5G networks. He has also worked on smart grids, information-centric networking, inter-domain carrier services and multicast/broadcast service provision over cellular networks. He was a Research Associate at Telecom ParisTech (France) and subsequently at the University College London (UK).

Vasileios Theodorou is a R&D Engineer at Intracom S.A. Telecom Solutions, Greece, working on NFV and Edge Computing. He has been a Research Associate at the Database Technologies and Information Management Group of the Polytechnic University of Catalonia (UPC), Spain and a Research Assistant at the Adaptive Systems Research Lab of York University of Toronto, Canada.

Sergios Soursos is a Master R&D Engineer at Intracom S.A. Telecom Solutions, Greece, working on 5G networks, IoT interoperability and Cloud/Edge Computing. He has also worked on cloud traffic management, information-centric networking, peer-to-peer networking and overlay network management.

Dimitrios Spatharakis is a PhD candidate in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. His research interests include cyber physical systems, IoT, NFV/SDN and Edge Computing.

Ioannis Dimolitsas is a PhD candidate in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. His research interests include multi-criteria optimization, fedetated clouds and NFV/SDN.

Marios Avgeris is a PhD candidate in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. His research interests include control theory, cloud computing, IoT, semantic web technologies and network monitoring.

Dimitrios Dechouniotis is a Senior Researcher in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. He has worked for many European and national research projects. His research interests include control theory, cloud computing and IoT.

Symeon Papavassiliou is a Professor in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. He has an established record of publications in the field of next generation network modeling, optimization and management, with more than 300 technical journal and conference published papers.