Edge Cloud Selection: The Essential Step for Network Service Marketplaces

Ioannis Dimolitsas*, Dimitrios Dechouniotis*, Symeon Papavassiliou*, Panagiotis Papadimitriou[†], Vasileios Theodorou[‡]

*National Technical University of Athens, Greece, {*jdimol, ddechou*}@*netmode.ntua.gr, papavass*@*mail.ntua.gr*

> [†]University of Macedonia, Greece, *papadimitriou@uom.edu.gr* [‡]Intracom Telecom, Greece, *theovas@intracom-telecom.com*

Abstract—The evolution of 5G network service ecosystems raises the need for the discovery of services among multiple (edge) clouds with potentially heterogeneous resources and features. Besides 5G service deployment, such a service discovery architecture could provide the means for cross-service communication (CSC), i.e., the consumption of a network service (NS) by another NS, empowering the service capabilities of the latter. This article presents a distributed service discovery framework for Network Service Marketplaces (NSM) in edge computing. We exemplify the operation of a new distributed service discovery mechanism that relies on a cache-based protocol for the discovery of NSes among edge clouds. We also present a multi-criteria ranking mechanism, which identifies the most suitable edge cloud according to the client's functional, performance, and cost requirements. Our evaluation results indicate that the proposed Service Discovery Mechanism (SDM) yields a service discovery effectiveness on par with other approaches, while drastically reducing the communication overhead.

INTRODUCTION

Innovative 5G verticals are composed of various time- and mission-critical applications that are developed as end-to-end services over heterogeneous distributed computing and network infrastructures. Towards network delay minimization and increased reliability and bandwidth savings, infrastructure providers offer computing, network and storage resources at the edge of the network. To this end, Edge Computing has become the emerging service delivery paradigm [1], raising various service management and orchestration challenges. The Edge Clouds (ECs) are small-sized computing facilities directly connected to a wireless access point. Under this setting, the dominant network architecture concepts of Network Function Virtualization (NFV) and Software-Defined Networking (SDN) continuously evolve to fulfill the various (non-)functional requirements for service orchestration of 5G verticals.

The deployment of 5G verticals in ECs is based on the concepts of network slicing and network services (NSes), which provide the desired isolation among tenants. However, the co-hosting of such service ecosystems creates opportunities for cross-service interactions within the same EC infrastructure, where a NS can consume a NS of a different tenant. The cross-service communication (CSC) enhances the overall performance in terms of throughput, latency and potentially other Key Performance Indicators (KPIs), and creates opportunities for financial rewards through service leasing. In a similar manner, service components may be shared among different service operators, *i.e.*, a service orchestration aspect, known as inter-service operability. These emerging trends pave the way towards a next-generation Network Service Marketplace (NSM), where users are able to compose custom service chains, while a service will be empowered to consume other co-located services in a controlled and secure manner [2]. The NSM should provide several automated functionalities to both service providers and consumers, while supporting the entire life-cycle of the NSes, including service registration, discovery, instantiation and termination. Furthermore, it must provide secure and automated mechanisms for the CSC establishment.

Every user can act either as service provider or consumer. Thus, the NSM should provide the essential mechanisms for service registration and discovery. The service registration should allow users to describe the features and requirements of their NSes in a consistent and open manner that can be used by the participating heterogeneous ECs. With this capacity, ETSI NFV data model describes analytically the structure and parameters of the network services [3]. This model can be further enriched with application descriptors which provide Quality of Service (QoS) requirements. Upon service registration, the service discovery empowers users to identify the appropriate NSes for CSC. A NSM consists of geographically dispersed ECs for maximized footprint, with each one locally maintaining information about the CSC-available NSes. The service discovery is performed either centralized or in a distributed manner. A centralized approach commonly entails scalability limitations, while the efficiency of a distributed discovery mechanism is strongly affected by the underlying EC infrastructure.

Under this complex setting, the ECs of a single provider may have heterogeneous resources, which implies different features (e.g., hardware acceleration, optimized CPU instruction sets, etc.), which, in turn can lead to performance differentiation between ECs. Furthermore, the resources of ECs are limited, while resource allocation must be carried out such that service KPIs are met. Thus, prior to service component (e.g., virtual network function) placement, the candidate ECs, assessed and proposed by the service discovery mechanism, should be evaluated based on multiple performance, cost, and support criteria [4]. Thereby, the final EC selection problem can be formulated as a multi-objective optimization problem, based on well-defined KPIs, which are common for every EC.

In this respect, we propose a distributed service discovery and multi-criteria EC selection mechanism for enabling 5G network slicing and CSC at the network edge. We assume that a service embedding request can be handed onto any EC, which is held responsible for the discovery and evaluation of candidate ECs, based on a multitude of criteria. The contribution of our work is twofold. First, we design a cache-based discovery protocol, which aims at discovering the required services for CSC with fast convergence and controlled communication overhead. Second, we employ a multi-criteria decision making (MCDM) approach to rank the candidate ECs based on the individual requirements of each service request. The distributed service discovery framework is evaluated using realistic EC-

level topologies.

In comparison with both centralized and distributed service discovery solutions (*i.e.*, flooding), our proposed approach yields significant reduction in the communication overhead, without any perceptible penalty in terms of discovery efficacy. Our evaluation results show that the proposed ranking mechanism is scalable with the number of ECs.

RELATED WORK

Recently several projects have focused on the establishment of NSM, which provide service registration and discovery functionalities, and enable crossservice communication. Aligned with ETSI NFV architecture [5], T-NOVA introduces the notion of the Network Function Store, which empowers clients to compose services out of individual network functions, supplied through this store [6]. Similarly, the FENDE Marketplace enables developers to onboard services as Virtual Network Functions (VNFs) in a service catalogue, whereas users can select among the available VNFs and plug them into an instantiated slice on the underlying infrastructure [7]. Based on the concept of Resource Broker, the NECOS Marketplace facilitates the resource discovery for instantiating network slices across multiple resource providers [8]. MESON advocates crossslice communication between co-located slices in a EC, with respect to the requirements of both slice tenants [2]. Based on open source MANO (OSM) [9], the MESON orchestration provides CSC-enabled service discovery, EC selection, and slice placement. Considering the distributed service discovery across multiple providers or Points-of-Presence (within a single provider), the 5GEx hierarchical architecture of multi-domain orchestrators enables the exchange of high-level information with clients, service discovery and mapping, as well as VNF configuration and monitoring [10].

DISTRIBUTED EC SELECTION MECHANISM

For the proposed edge cloud selection approach, we consider an edge cloud infrastructure comprising geographically dispersed and heterogeneous ECs, as shown in Figure 1. Furthermore, the ECs are clustered in availability zones within specific areas, while each node in Figure 1 represents an EC. Each EC's resources are controlled by a Virtual Infrastructure Manager (VIM), which instantiates the services and assigns resources to them. A new



Fig. 1: Distributed Service Discovery.

network service embedding (NSE) request can be submitted to any EC, which selects the appropriate EC for hosting the NS. For the final selection, various functional, Quality of Service (QoS), and cost requirements, as well as a set of desired CSCenabled services are taken into account. Within each EC, several components are deployed to realize the service discovery and EC selection. In particular, the services of the NSM are stored in the Service Catalogue of each EC, whereas the Service Cache contains the available CSC services available in the adjacent ECs. The Service Discovery Mechanism (SDM) is responsible for sending query messages to its neighbors for CSC service discovery. Finally, based on a MCDM method, the Edge Cloud Ranking Mechanism (ECRM) evaluates the candidate ECs and selects the most appropriate one for NS deployment. The following sections describe in detail the functionality of the above components.

Service Catalogue and Service Cache

The Service Catalogue and Service Cache are the key components of the NSM, containing all essential information of the network services that are available for CSC. In particular, each EC holds a Service Catalogue, which is a list of CSC-enabled service instances deployed at its edge infrastructure. Following the MESON architecture, the service specification is described via ETSI NFV network service descriptors and extended ETSI MEC Application descriptors [2]. Based on this information, the matching between the requested services and the available CSC service instances is performed.

Additionally to the Service Catalogue, the EC's Service Cache holds the most recent information

about the available CSC-enabled services of the neighboring ECs. As shown in Table I, for each entry, the Service Cache has fixed size and FIFObased updating policy.

Service Discovery Mechanism

The Service Discovery Mechanism is responsible for the query message forwarding to discover the ECs that provide the desired CSC-enabled services. The message forwarding is based on the ECs' Service Cache content, which contains the *id* of the EC neighbour with the most recent advertised CSC-enabled services. The message contains the requested CSC-enabled services and information about the forwarding policy, such as the previously traversed ECs and the TTL value.

A NSE request can be submitted to any EC, which is termed as *Search Node*, and the forwarding policy is based on the cache content of the *Search* Node. If one or more desired CSC-enabled services match with one or more cache entries, the query message is forwarded to these ECs. Otherwise, the Search Node forwards the message to all of its neighbors. In both cases, TTL is decremented by one. These forwarding rules are applied to any EC that receives a query message, till TTL becomes zero. As mentioned, the already traversed ECs are encapsulated in the query message to avoid repetitive messages on the same nodes. Also, each EC that has received a query, responds to the sender EC with its most recent CSC-enabled services to update the respective cache entry. Finally, whenever at least one service match occurs, the EC sends a response to the Search Node, which contains information about the EC evaluation in the ECRM component.

Figure 1 illustrates an example of the SDM with three availability zones, namely *Zone 1, Zone 2, Zone 3.* Table I shows the cache entries of the ECs (2.a), (1.a), (1.c), (2.b) and (2.c). For this example, the TTL value is set to 3 and each cache entry contains up to three services for each neighbour EC. A NSE request is submitted to EC (2.a) (*Search Node*) that requires the CSC-enabled service s1. This service is hosted in the (1.a), (1.b), (2.b), (3.a) and (3.d) ECs. Based on its cache entries, the *Search Node* (2.a), forwards the request to the (1.a) and (2.b) ECs, as the s1 matches with its corresponding entries for these ECs. Also, the (1.a) and (2.c) ECs forward the request based on the s1 record in their caches. The ECs that are flooding the request are

EC (2.a) Cache	EC (1.a) Cache	EC (1.c) Cache	EC (2.b) Cache	EC (2.c) Cache
(2.b) - [s1, s4, s17]	(1.c) - [s16, s2, s1]	(1.a) - [s6, s14, s9]	(2.a) - [s5, s20, s7]	(2.a) - [s2, s15, -]
(2.c) - [s5, s8, s2]	(1.b) - [s3, s18, -]	(1.b) - [s3, s5, -]	(2.c) - [s2, s17, s5]	(2.b) - [s3, s11, s5]
(2.d) - [s2, s9, -]	(2.a) - [s4, s6, s12]	(3.a) - [s8, s12, s4]	-	(2.d) - [s7, s12, s6]
(1.a) - [s1, s6, s14]	-	-	-	(3.d) - [s3, s1, s8]

TABLE I: EC Service Cache Example.

the (1.c) and (2.b), as their caches have no matching entry for s1. As shown in Table I, the entry in the cache of (1.a) is out of date, as the s1 is no longer hosted in the EC (1.c). When the ECs (1.b), (3.a) and (3.d) are reached, they do not forward further the request, as the TTL has zero value, so their caches are not being included in the Table I. Every traversed EC sends two response messages. The first one is an update message to the neighbour EC, which has sent the request, and contains the three most recent CSC-enabled services to be cached. The second is a response message to the Search Node about the requested service(s). The ECs, hosting the requested service(s), are the candidate ECs and include to their responses to the Search Node the necessary information for the evaluation process.

Edge Cloud Ranking Mechanism

A NSE request can be submitted to any EC, which has a dedicated functionality for the edge cloud selection. In each EC, the ECRM selects the most appropriate EC for NS deployment and CSC establishment based on a set of functional, performance and cost criteria and the set of desired CSC-enabled services. Extending the ranking mechanism of [11], the proposed ECRM is based on Analytic Hierarchy Process (AHP), which handles various types of numerical criteria. This provides the following enhancements: (i) the hard requirements are included in the structure of AHP in order to provide a more fine-grained evaluation regarding the hard requirements of the NSE request, (ii) the Virtual Edge Cloud (VEC) is defined to express the set of user's hard and soft requirements, in terms of ranking data. VEC profiles are defined based on the weight assignments and are used as a point of reference in the ranking process. For the technical details of AHP, the interesting reader may refer to [11]. In the following, for completeness, we provide an overview of the ECRM.

Ranking Criteria - Hierarchical Structure. The AHP is based on a hierarchical structure, which consists of (technical) KPIs and (non-technical) attributes, as shown in Figure 2. In our approach, the

ranking is based on two sets of service requirements, namely *Hard Requirements* and *Soft Requirements*. Contrary to a simple filtering solution, the hard requirements (*i.e.*, the *Availability Zone* of the EC and the *CSC-enabled Services*) are included in the AHP model. This allows the ECRM to evaluate more EC candidates, which can be in different availability zones of the user's request. On the other hand, the *Soft Requirements* refer to performance, cost and technical support criteria [11].

Weight Assignment - User's Requirements. As shown in Figure 2, the weights on the edges of the hierarchical structure correspond to the relative importance of each KPI and attribute in the ranking computation and represent the individual requirements of every service request. Towards the customization of the user's requirements, the ECRM allows the user to assign the corresponding weight values for specific attributes and KPIs. In particular, the weights between the Hard Requirements, the Soft Requirements, the Performance and their descendant are user-defined. For each group of siblings attributes or KPIs, the sum of their weight values must be equal to one. Users are not allowed to define other weights in the structure in order to avoid misconceptions in the ranking process. Thus, a Consistency Ratio is computed for each group of sibling KPIs or attributes to avoid inconsistencies in the weight assignments [11].

AHP Ranking. After the discovery process, each candidate EC, in its response message to the *Search Node*, advertises its data for the corresponding KPIs of the hierarchical structure, which are taken into account in the ranking process. At any level of the hierarchy, for each attribute, a Relative Ranking Vector is computed, which provides a normalized score for every candidate EC for the specific attribute. Starting from the bottom level of the hierarchy, the upper-level vectors are computed. The top-level attribute corresponds to the overall ranking of the ECs. Eventually, the highest-scored EC is selected for the service deployment.

VEC Candidate. Since the AHP provides a relative



Fig. 2: Hierarchical Model of Ranking Mechanism.

ranking of the EC candidates, we introduce the concept of the VEC, as point of reference, which quantifies the distance between the user's requirements and the candidate EC' offerings. Therefore, the appropriate values in the respective KPIs must be defined according to the user's requirements. So, based on the user-defined weight assignments, a specific VEC profile is selected automatically. Using k-means algorithm and datasets of alternative weight assignments and KPI values, we define various VEC profiles. The cluster's parameters are the weight values of the attributes Cost (relative to the Performance), Computing Performance and Network Performance, which prioritize the importance between the competitive criteria. As such, six different clusters of VEC are computed: Low Cost - High Computing Performance, Low Cost-High Network Performance, Low Cost - General Performance, Indifferent Cost – General Performance, Indifferent Cost – High Computing Performance, and Indifferent Cost – High Network Performance. Each VEC profile has predefined values for the KPIs to better reflect the user's requirements. Especially, the KPIs of the Hard Requirements are set equal to the user's desired values. The VEC is evaluated as a candidate EC and its relative score implies that ECs with higher ranking satisfy the user requirements.

EVALUATION

In this section, we evaluate the efficiency of the proposed edge cloud selection mechanism. To this end, we consider a network of 300 ECs, subdivided into three availability zones (*Zone1*, *Zone2*,

and Zone3). Based on real network topologies of regional providers [12], the size of these availability zones is set to 50, 100 and 150 ECs respectively. A pool of 20 services for enabling CSC is available at every EC. Each EC can host from 5 to 15 CSC-enabled services. Furthermore, each EC advertises its values on the KPIs of the AHP model. These values are randomly selected using a uniform distribution. In order to highlight the impact of the important system parameters on the results of discovery and ranking, TTL varies from 1 to 8, while the size of Service Cache ranges from 3 to 5. In the following experiments, for each TTL value, 90 NSE requests with random values of hard and soft requirements are generated and uniformly assigned to the ECs of the three availability zones. Each of those requests is being handled by the proposed mechanism for the different cache sizes. Furthermore, one or two CSC-enabled services are required in every NSE request.

The first experiment aims at assessing the SDM efficiency. In particular, the SDM is compared against a standard discovery technique, termed as Flooding, which broadcasts the query message to every neighbour of the sender EC. Flooding relies on the Breadth-First-Search (BFS) approach [13], which prevents repeating queries on an EC. In the following, we compare these two techniques in terms of generated communication overhead and network utilization. Also, we compare our solution with the NECOS centralized approach for discovery in cloud resource marketplace [8], in terms of the total number of exchanged messages. The NECOS centralized broker is responsible for sending the NSE request in every EC and processing the corresponding responses.

Regarding the ECRM evaluation, we demonstrate a ranking of three ECs in order to quantify the potential gains from the use of the Hard Requirements in the AHP (instead of employing a simple filtering solution). We further assess the influence of the VEC in the ranking results. Finally, we provide additional remarks about the overall performance of the proposed mechanism.

SDM Evaluation

In this experiment, we compare the efficiency of the SDM against the *Flooding* technique with diverse values of the TTL and the Service Cache size. In this respect, we measure the communication



Fig. 3: Generated Communication Overhead.

overhead as the percentage of utilized links. As shown in Figure 3, the proposed SDM severely reduces the utilized links in the search-tree, compared to *Flooding*. More precisely, SDM yields a communication overhead of 9.5% on average, whereas the respective value for *Flooding* is 23.2%. This low communication overhead is also reflected by the number of messages exchanged. In particular, the SDM generates 138 messages, as opposed to Flooding at which 220 messages are exchanged (i.e., 38% reduction for SDM). We note that, in the SDM, each EC generates two messages, the cache update message and the response to the Search Node, while the Flooding approach generates only one response message. Regarding the comparison with NECOS, we conduct experiments by setting the TTL equal to 8 to measure the generated messages in a search depth equal to the centralized approach. For different cache sizes (3 to 8), 30 requests are submitted from each of the availability zones. The centralized broker approach overall requires 600 request/response messages, while SDM generates an average of 304 messages, which further corroborates the efficiency of SDM compared with centralized solutions in terms of communication overhead.

Figure 4 illustrates the discovery capability of SDM (with a range of TTL values and cache sizes) and *Flooding*. We use the *Cache Hit Rate* as metric, which corresponds to the ratio of the discovered ECs with at least one required CSC-enabled service over the total number of queried ECs. The average *Cache Hit Rate* for the SDM and *Flooding*, across all conducted tests, are 56% and 47%, respectively. Furthermore, 76% of the candidate ECs evaluated by the ECRM are in the availability zone of the *Search*



Fig. 4: Cache Hit Rate.

Node. Due to more queried ECs, this percentage is lower for the *Flooding* approach. Regarding the cache size, the larger the cache is the hit rate is slightly better (2 - 3%). From our tests, we infer that an efficient cache size is approximately 20% of the total available CSC-enabled services.

ECRM Evaluation

The first two columns of Table II include the name, the weight, and the type of KPIs of the AHP model. The following three columns contain the values of these KPIs for three ECs, whereas the last column contains the KPI values of the VEC, according to the chosen profile and the user's hard requirements. Following a bottom-up approach, a ranking vector is computed for the ECs, including the VEC. Due to lack of space, the rest of the attributes of the upper levels are omitted. All attributes and values of the weights are shown in Figure 2.

As a point of reference, the VEC profile of *Indifferent Cost* – *High Computing Performance* is used in the ranking process. Following the AHP method, the overall ranking result is: [0.322, 0.244, 0.196, 0.236], which means that the EC ranking from the best to the worst is: (1) EC1, (2) EC2, (3) VEC, (4) EC3. As such, the EC3 is considered as a not acceptable solution, since it is ranked below the VEC. The inclusion of the hard requirements in the AHP model and the addition of the VEC profile in the evaluation lead to a more fine-grained ranking, promoting the candidate ECs that better satisfy the user's requirements.

Also a comment on the overall performance can be made. For all experiments, evaluating the candidates ECs discovered by the *Flooding* technique

KPIs	Metric	EC1	EC2	EC3	VEC
Availability Zone (0.6)	boolean	1	1	0	1
CSC-enabled Ser. (0.4)	#	2	1	2	1
Serv. Availability (0.4)	%	95	85	85	90
Service Response (0.3)	ms	15	20	15	10
Elasticity (0.3)	# VMs	3	1	2	2
Bandwidth (1.0)	Mbps	20	30	30	20
Data Cost (0.3)	\$	0.3	0.25	0.25	0.5
VM Cost (0.7)	\$	5	15	10	20

TABLE II: Advertised EC KPI Data - VEC Data

provides at least one solution with higher ranking than the VEC. For the candidate ECs discovered by the SDM, the ECRM provides at least one solution that satisfies the user's requirements for 99.5% of the experiments. This result implies that most of the selected ECs for NS deployment are within the availability zone of the *Search Node* and that the proposed mechanism achieves identical efficiency with the greedy *Flooding* approach with significantly lower communication overhead.

CONCLUSIONS

Enabling Network Service Marketplaces is crucial for reaping the benefits of 5G network technologies. To this end, efficient CSC at the network edge can pave the way for service provisioning and cross-service interactions. In this article, we presented a distributed EC selection framework towards 5G Network Service Marketplaces. The proposed framework encompasses: (i) a service discovery mechanism, which yields low communication overhead and fast convergence, stemming from caching and the efficient reduction of search space, and (ii) a multi-criteria ranking mechanism, which evaluates candidate ECs in terms of service deployment.

Our future work will focus on creating more sophisticated VEC profiles through Machine Learning techniques to better represent the individual requirements of the NSE requests. Also, instead of selecting a single EC for service deployment, we will extend our solution for NS embedding across multiple domains in a distributed manner.

ACKNOWLEDGMENTS

This work is supported by the MESON (Optimized Edge Slice Orchestration) project, cofinanced by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE - INNOVATE (project code: T1EDK-02947).

REFERENCES

- D. Dechouniotis, N. Athanasopoulos *et al.*, "Edge Computing Resource Allocation for Dynamic Networks: The DRUID-NET Vision and Perspective," *Sensors*, vol. 20, no. 8, p. 2191, 2020.
- [2] G. Papathanail, A. Pentelas *et al.*, "MESON: Optimized Cross-Slice Communication for Edge Computing," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 23–28, 2020.
- [3] ETSI GS NFV-SOL 006, "Network Functions Virtualisation (NFV); Protocols and Data Models," 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/006/02.06.01_60/gs_NFV-SOL006v020601p.pdf
- [4] R. V. Rosa and C. E. Rothenberg, "The Pandora of Network Slicing: A Multicriteria Analysis," *Trans. on Emerging Telecom. Tech.*, vol. 31, no. 1, pp. 1–17, 2020.
- [5] ETSI GS NFV 002, "Network Functions Virtualisation (NFV); Architectural Framework," 2013. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01 _60/gs_NFV002v010101p.pdf
- [6] M.-A. Kourtis, M. J. McGrath *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, 2017.
- [7] L. Bondan, M. F. Franco *et al.*, "FENDE: marketplace-based distribution, execution, and life cycle management of VNFs," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 13–19, 2019.
- [8] S. Clayman, A. V. Neto *et al.*, "The NECOS Approach to Endto-End Cloud-Network Slicing as a Service," *IEEE Communications Magazine*, 2021.
- [9] ETSI, "Open Source MANO." [Online]. Available: http://osm.etsi.org
- [10] R. Guerzoni, I. Vaishnavi *et al.*, "Analysis of end-to-end multidomain management and orchestration frameworks for software defined infrastructures: an architectural survey," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, pp. 1–19, 2017.
- [11] I. Dimolitsas, D. Dechouniotis *et al.*, "A multi-criteria decision making method for network slice edge infrastructure selection," in 2020 6th IEEE Conference on Network Softwarization (Net-Soft). IEEE, 2020, pp. 1–7.
- [12] The University of Adelaide, "The Internet Topology Zoo." [Online]. Available: http://www.topology-zoo.org/index.html
- [13] T. H. Cormen, C. E. Leiserson *et al.*, *Introduction to Algorithms*. MIT press, 2009.

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, federated clouds, edge computing and NFV/SDN.

Dimitrios Dechouniotis is a Senior Researcher in the School of Electrical and Computer Engineering (ECE) at National Technical University of Athens. 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.

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 and edge computing.

Vasileios Theodorou is a Senior R&D Engineer at Intracom S.A. Telecom Solutions, Greece, working on NFV and Edge Computing.