A survey on distributed NFV multi-domain orchestration from an algorithmic functional perspective

Josué Castañeda Cisneros, Sami Yangui, Saúl Eduardo Pomares Hernández, Khalil Drira

A survey on distributed NFV multi-domain orchestration from an algorithmic functional perspective

Josué Castañeda Cisneros*, Sami Yangui*§, Saul E. Pomares Hernández*†, Khalil Drira*
*LAAS-CNRS, Université de Toulouse, §INSA, F31400, Toulouse, France. † INAOE, 72840, Santa María Tonantzintla,Puebla, Mexico. {jcastane, yangui, drira}@laas.fr, spomares@inaoep.mx

***Abstract*—Under the Network Function Virtualization multi-domain orchestration approach, many service providers jointly manage the lifecycle of network services composed by Virtual Network Functions (VNFs). Many orchestration algorithms have been published focusing on different tasks for the lifecycle of network services. Nowadays, some general-purpose and architectural orchestration surveys have been published. However, currently, no survey classifies and evaluates distributed multi-domain orchestration algorithms. In this paper, we focus on multi-domain orchestration works from an algorithm perspective by considering the cooperative and competitive approaches. Algorithms are classified and evaluated. We propose a taxonomy for works that consider different lifecycle tasks of network services. Classification and evaluation are done by identifying key requirements present in competitive and cooperative scenarios. This allows us to have a perspective on the current state of multi-domain orchestration algorithms. We also identify future research directions based on such a current perspective.**

***Index Terms*—Multi-domain Orchestration, Close Federations, Network Function Virtualization**

I. Introduction

With softwarized networks, service providers reduce their capital and operational costs of network services, as they run on generic hardware [1]. The Network Function Virtualization (NFV) ensures compatibility between different providers [1]. This allows many orchestrators to jointly provision and manage services. Such a paradigm is known as multi-domain orchestration (MdO).

Sharing resources allows providers to lower even more their costs and extend their market reach. However, MdO changes the way services are provisioned compared to the traditional single-domain paradigm [2]. In single-domain orchestration, the orchestrator has a global view of the system [1]. In MdO, the orchestrators have only local information (i.e. they do not know the resources and topologies) [2]. To provide consistent services, the orchestrators exchange information [3]. Many works focused on MdO have been proposed [2]; however, no work offers a broad perspective on the state of the art of MdO algorithms.

Current survey papers, focus on service orchestration [2], or consider the evolution of architectural designs [3]. No published survey considers only MdO algorithms. Thus, there is no perspective on the algorithm's state, nor the possible research directions of the field. This work fills the gap in the literature by considering MdO algorithms.

We create a taxonomy of NFV MdO algorithms. We classify works based on their coordination approach. After, we identify key requirements for MdO algorithms, such as profitability, with two use cases published in the literature. Based on these requirements we evaluate the works and describe the state of current algorithms. Then, we offer perspectives for research directions in MdO algorithms. Our major contributions are the classification and review of the literature, and the identification of promising research directions for distributed MdO algorithms.

This paper is organized as follows. Section II describes concepts related to MdO and how the relevant work's classification was made. Section III presents the requirements needed for a successful orchestration algorithm. Next, Section IV presents a brief review of the considered works. Section V contains the main findings after contrasting the works with the requirements previously identified. It also describes the research direction based on the findings. Section VI concludes the paper.

II. Design Considerations and Literature Classification

Under NFV, the network services are provisioned by Virtual Network Functions (VNFs). The VNF lifecycle is inspired by the service provisioning lifecycle detailed in SOA [1]. Figure 1 depicts the operations that make up a typical network service's lifecycle. First, the service provider models the network components: resource requirements, connections, and scripts to execute. Then, the providers publish their VNFs. After, a provider allocates resources. Next, the provider selects how multiple components should connect creating a complex service. Finally, by monitoring the service, the provider can reconfigure the network component.

We present MdO by a three-layered schema, as shown in Figure 1. The first layer, at the bottom, describes the ordered tasks to deploy services. The second layer, in the middle, contains multiple orchestrators. The third layer, on top, encompasses previous layers with specific properties for MdO.

The top layer of Figure 1 covers all the previous layers. The challenges of MdO come from the **limited information** among orchestrators. Limited information enables privacy and autonomy for providers, increasing flexibility; at the cost

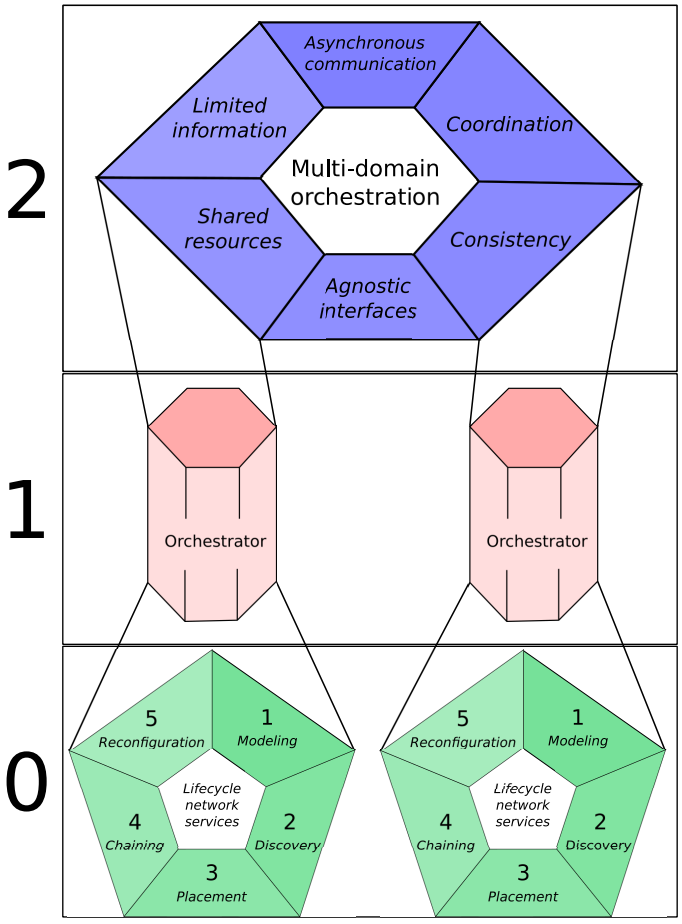

Fig. 1. Multi-domain orchestration view in layers. The bottom is the lifecycle of network services. The middle consists of multiple orchestrators from different providers. The upper has all the added constraints of multi-domain orchestration.

of greater overhead in terms of computational complexity. For example, when orchestrators keep their information private, more redundant messages and orchestration tasks are required to deploy a shared network service. This trade-off reflects on other challenges, as they are related. First, since the orchestrators communicate via non-deterministic channels (i.e. communication channels where messages can be lost and arrive at any time, in any order), orchestrators need to handle **asynchronous communication** for all tasks in the bottom layer. As providers **share services**, they need to **coordinate** to prevent unwanted side-effects while executing a task in the service's lifecycle. To remove any conflicts between the orchestrators, the providers must ensure **consistency** for services. **Agnostic interfaces** enable the orchestrators to communicate and share the information. This disseminates knowledge between them enabling consistent provisioning. Next, we describe the literature classification.

A. Literature classification

We surveyed the literature on NFV MdO works by reviewing papers published from 2014 to 2021. We use as keywords *Multi-domain orchestration federation*, *Multi-domain orchestration*, *multi-domain NFV*, *multi-domain VNF*, *cross-domain orchestration*, and *multi-domain SFC*. After reading and validating the corpus, we organized it into two categories: architecture proposals or algorithmic solutions. Architecture proposals extend the current ETSI architecture for a single domain and propose different interfaces to meet the goals of MdO. Algorithmic solutions focus on solving orchestration problems related to the service's lifecycle tasks. We focused on algorithmic solutions, as we could measure objectively the advances in MdO and future research directions. Relevant algorithmic papers were organized in the taxonomy shown in Figure 2.

NFV MdO is on the taxonomy's root branch (Figure 2, first level **I**). This orchestration type involves multiple administrative domains, managed by different service providers, that jointly offer network services. Depending on how the orchestrators behave, they can be centralized, hierarchical, or distributed (Figure 2, second level **II**). Centralized orchestration has a single global orchestrator managing all lifecycle management of network services of all other providers. Hierarchical orchestration adds structures where local orchestrators manage each of their domains, reducing exposure and communication overhead. The global orchestrator on top of the structure is the mediator in case of conflicts when provisioning joint services. Distributed orchestration proposes the most flexible model to achieve each task of the lifecycle management of network services. All orchestration is done locally without a global reference of the federation. This ensures horizontal integration models, so new participants can join the federation while improving privacy among participants for security and economic reasons. A lot of works on the literature focus on the distributed branch as it addresses the security, privacy, and flexibility concerns promised in the goals of MdO. Thus, we focus on the distributed branch exclusively.

The literature on distributed orchestration considers two main branches: Open and close federations (Figure 2, third level **III**). Open ones foster competition among service providers as new operators can enter at any time. Close ones assume a fixed number of trusted participants known beforehand. The main divide among them is the trust of providers. Figure 2 shows how the taxonomy's branches differ. Open federations consider competitive solutions since orchestrators have byzantine behavior (e.g. one orchestrator sends inaccurate information to other orchestrators). Close federations assume trustful orchestrators (e.g. orchestrators always send accurate information) that enable both competitive and cooperative solutions (Figure 2, fourth level **IV**). Solutions in each branch are different. In cooperative scenarios, orchestrators communicate directly and exchange key information like the topology, resource, and service information using a unified communication model. In competitive scenarios, orchestrators do not have direct communication and must learn key information through others' actions, usually by means of indirect communications. Such communication mechanisms, found in the competitive works, include auctions and reinforcement learning. However, most works consider that there exists an orchestrator-to-orchestrator standard communication interface; however, this still is an open problem [2].

Algorithmic solutions consider static or dynamic approaches

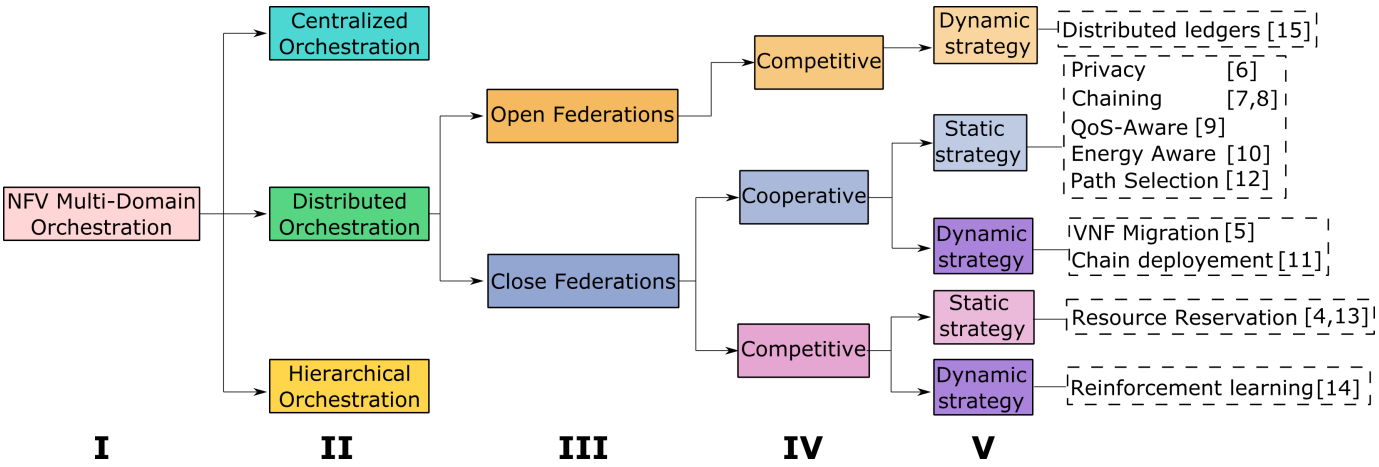

Fig. 2. NFV multi-domain orchestration taxonomy

(Figure 2, fifth level **V**). Static solutions consider a single-time provision; while dynamic ones enable solutions to adapt over time.

This survey focuses on the distributed orchestration under close federations as most of the work found in the literature for MdO is positioned under these categories. We discuss open federations for future perspectives and research directions.

III. Requirements

After reviewing the works, we identify nine requirements for successful distributed MdO algorithms and illustrate them based on two use cases. The first case **U1** is a competitive Massively Multiplayer Online Game running with many orchestrators [4]. Many concurrent players meet around a shared virtual environment. Specifically, the game service is deployed as a VNF chain, where each VNF offers specific game-play tasks such as graphics rendering, game physics, and character management. The second use case **U2** is a cooperative sharing monitoring scheme to identify valuable information [5]. It implements a city monitoring system where multiple providers share network services to handle citizens' various requests. Specifically, a content delivery network service is deployed using shared VNFs, such as a VNF Mixer.

- (R1) Support **interoperability** among the federation. This means network services can be built and interact using different VNFs from different administrative domains. This can be done by using standard interfaces and open APIs that comply with the multi-domain specification. For example, in **U1**, gamers have different machines with different requirements to play, such as personal computers, and phones. As new services can be added, the services must still be able to interact with the deployed ones.
- (R2) Deliver maximum **profitability** to providers for the end-to-end service's lifecycle tasks. This means not only reducing the capital and operational expenses but also achieving higher market value through the federation. Achieving profitability requires adapting to the changes in the environment to capitalize on new opportunities, but making sure such adaptation does not have unwanted effects. For example, in **U2**, the migration of the shared VNFs should minimize the latency for all affected services, not only for a single VNF. This usually involves finding the optimum with conflicting goals.
- (R3) Maintain **secrecy** of the business knowledge and detailed network information among providers. Secrecy is the guarantee that key information from an administrative domain is private and will not be disclosed to any other provider. To achieve such a requirement, only local orchestration is allowed and security for every VNF must be enforced. For example, in **U1**, providers do not share detailed information, such as topology, since they compete against other providers for resources.
- (R4) Ensure **accountability** among all providers. Accountability means that, in case of a problem, the orchestration solution could identify the degree of responsibility of each provider and ensure the SLA is satisfied. For example, in **U2**, suppose there is a problem with the last service delivered to a load balancer. The first step is to identify that the VNF Mixer is the culprit. Then, since the VNF is shared among two domains, the degree of responsibility must be established. Finally, according to a common SLA, the orchestrator is billed correspondingly.
- (R5) Maintain **consistency/availability** for all the network service's lifecycle tasks across the federation. Consistency and availability mean that the services maintain their functionality with their given properties despite reconfigurations on the fly. This can be done by coordinating the orchestrators or resolving conflicts automatically. For example, in **U2**, the migration algorithm must maintain the state of the shared Mixer VNF while still forwarding and processing the video being sent by multiple users.
- (R6) Enable **flexibility** to contextual changes across the federation. This means the services can be extended to support more complex functionality, users, and legacy services. Such requirements can be achieved by ensuring that algorithmic solutions adapt to new conditions in the environment by jointly monitoring services. For example, in **U1**, the game provider's algorithm re-configures

services to reflect the changes in the topology of each provider. For example, if there is a surge during a special release, the game provider instantiates regional VNFs to maintain the user's quality of service.

- (R7) Ensure **composability** of a logical request for each lifecycle task across the federation. Composability means the network services are always provisioned and managed by over one administrative domain. This allows the federation's ecosystem to extend more in future federation-to-federation orchestration. For example, in **U2**, the migration of the shared VNF takes place in the two domains where orchestrators must place and instantiate the new VNF. Otherwise, instantiation would only be isolated in a single domain.
- (R8) Enable **portability** of VNFs and network services between providers in the federation. This means that all components from a network service can be migrated to other administrative domains or federations without updating any component in the network service. To achieve portability, the orchestration solutions must abstract all the network service's lifecycle tasks, ensure long-term support for each component of a network service, and adhere to open standards. For example, in **U2**, the domain that migrates the faulty Mixer VNF should do it automatically without conflicts or manual configuration.
- (R9) Enforce **scalability** when executing the lifecycle tasks of network services. This means that the computation required, including the overhead, must grow below linearly. For example, in **U1**, when choosing a VNF candidate to migrate, the orchestrators send messages to select the appropriate candidate. Since many optimal solutions belong to the NP-hard problem class, we expect scalability will be fulfilled with heuristic algorithms.

The synergy between the nine requirements ensures the success of distributed MdO. For example, portability and composability work towards interoperability; while flexibility, consistency, scalability, and accountability bring profitability closer for all providers, etc. Despite this synergy, works in the literature lack one or more of the requirements. Next, we review and evaluate the considered corpus.

IV. Literature Review

The family of distributed MdO works under close federations can be organized in two blocks: cooperative and competitive environments. Both blocks share common properties, albeit meeting requirements in different ways. Next, we describe each work reviewing it based on the nine criteria identified.

A. Cooperative works

The migration of shared VNFs without global knowledge has been explored in [5]. The authors propose an algorithm to coordinate multiple orchestrators satisfying the SLA requirements and keeping the state of VNF. This prevents unwanted effects while migrating shared VNFs. The time complexity is $O(n^2)$ where n is the number of VNFs.

A lightweight and privacy-aware orchestration algorithm was proposed in [6]. The algorithm abstracts publicly available information, such as the aggregated network topology and the type of VNFs, to support a binary query-response for deploying a service function chain. Aside from the algorithm, the authors also quantify and evaluate the gain of privacy preservation for multi-domain environments. The proposed algorithm learns over time the best domains to support a service function chain. The time complexity of the algorithm is $O(n^3)$, where n is the number of candidate paths.

A distributed and local-only mapping algorithm for service function chains was studied in [7]. The vertex-centric algorithm computes all feasible mappings and prunes them to get the optimal solution while satisfying a service's constraint and policies. For fixed-order requests, the number of chains grows exponentially. For flexible-order requests, where the chains can change, the number increases even more.

The service function chaining provisioning problem with a two-tier cloud network was proposed in [8]. The goal is to select and connect VNFs after a user creates a request; thus, minimizing the cost for the user. The authors model the throughput maximization problem. Since requests may share the same VNF type for the service chain, the authors proposed a heuristic algorithm. It provides an approximation ratio for a special case of the problem without end-to-end delay requirements. The algorithm's time complexity is $O(n^5)$ where n is the number of cloudlets.

Deployment of VNFs using SDN has been addressed in [9]. The authors propose an algorithm to optimize the energy consumption during the deployment of VNFs by using different variants of evolutionary algorithms. Their proposed algorithm got better results than traditional genetic algorithms; yet, their worst-case scenario still has a time complexity of $O(mn \log n)$ where m is the number of independent comparisons and n is the number of VNFs.

A model to solve the adaptive and dynamic VNF-FG considering migration of VNFs was studied in [10]. The authors propose a decentralized and heuristic algorithm. It allows VNF-FGs and VNFs to reallocate dynamically. The algorithm reallocates the VNF-FGs by cooperating domains that keep private information. The algorithm can optimize network utilization while limiting the number of migrations of VNFs. The time complexity is $O(mnp)$ where m is the number of virtual links, n is the number of mappings of virtual edges to substrate nodes and links, and p is the number of acyclic paths.

The network service chain's deployment across multiple domains is studied in [11]. A multi-objective optimization model and a heuristic algorithm were proposed. The algorithm considers the dependency relations among the VNFs and uses the Dijkstra algorithm to further reduce delay and cost for the overall chain. The time complexity is $O(mn)$ where m is the number of service requests, and n is the number of VNFs for each chain.

A horizontal-based service chaining algorithm was proposed in [12]. The heuristic algorithm maps the chains to the infrastructures. This is achieved by different orchestrators coordinating and selecting the best candidates. Such an approach

gives better results compared to an uncooperative method. The time complexity is $O(mn)$ where m is the number of domains, and n is the number of paths.

B. Competitive works

Cross-domain interaction among multiple providers was studied in [4]. The authors propose an auction-based market where service providers exchange network resources by buyer/seller transactions. They propose a distributed multi-agent deep reinforcement learning approach to win auctions. The algorithm performs well in dynamic environments; however, it down performs when collusion among agents is present. To determine the winner, the authors leverage on a $O(E+V)$ depth-first search on the bid graph G, where E is the set of edges and V is the set of vertices.

The allocation of the VNF-Forwarding graph was studied in [13]. The authors propose a deep reinforcement learning algorithm to place VNFs and Virtual Links where network operators hide their infrastructure in other competing domains. The algorithm gets better results in non-cooperative domains by reducing bidding prices while improving the deployment of VNFs. The time complexity is $O(nm \log m)$ where n is the number of VNFs and m is the number of paths.

A reinforcement learning algorithm was explored in [14]. The orchestrators learn how to slice service provisioning requests to increase revenue. Such an approach allows for dynamic decision-making without compromising privacy among orchestrators. The time complexity for the training algorithm is $O(mnpq)$ where m is the iterations, n the requests, p the episodes, and q the time-steps.

C. Review Synthesis

Three values are assigned to each work according to the degree they meet the criteria: Yes (Y), No (N), and Partially (P). Based on our evaluation, we discarded three criteria **(R1, R7, R8)**. **R1, R7** were removed as they were satisfied by all works; conversely, **R8** also was removed as no work satisfies it. The evaluation of the rest of criteria is summarized in Table I.

TABLE I
REVIEW SYNTHESIS OF CONSIDERED WORKS.

	Work										
	Cooperative								Competitive		
Requirement	5	6	7	8	9	10	11	12	4	13	14
2	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y
3	N	N	Y	N	N	N	N	N	N	N	Y
4	N	Y	N	Y	Y	N	N	Y	P	Y	N
5	Y	Y	N	P	N	N	P	N	N	P	N
6	Y	Y	P	N	Y	N	Y	N	Y	N	Y
9	N	N	N	N	P	N	P	P	P	P	N

Profitability (**R2**) is the most satisfied criteria by works (≈73%). Next, accountability (**R4**) and flexibility (**R6**) criteria are tied to the number of works that satisfy them (≈50 - 59%). They are related as contextual contingencies change network services, and in case of an error, the federation can determine who handles what. Consistency (**R5**) is barely met (≈32%). This suggests that most works on closed federations do not handle conflicts among the orchestrators during the lifecycle management of network services. Finally, secrecy (**R3**) and scalability (**R9**) are the least met criteria (≈18 - 22%) as only one work considers it.

The previous numbers show the current status of the literature for MdO algorithms. First, although all the works consider the interoperability criteria (**R1**), there is currently no standard for communicating orchestrators. Second, when seeing the composability criteria (**R7**) alone, all the works seem to comply fully with it. However, these criteria relate with the flexibility (**R6**) and consistency (**R5**), as orchestrators share network services. The contrast between the highest and lowest criteria met raises a concern. This can be explained by the network and orchestrations' assumptions for works when they share resources. For example, works consider a zero-latency network and total knowledge for the orchestrators via synchronization protocols. When such unrealistic assumptions are discarded, the orchestration algorithm shows its lack of performance. Third, although profitability (**R2**) is mostly satisfied, the lack of consideration for portability and consistency hinder meeting the goal of profit. For example, considering only, cost, energy, or latency suffices for the current user's needs. However, shortly, privacy constraints will come to the foreground; yet, most works in MdO consider it. Fourthly, despite accountability (**R4**) being satisfied by more than half the works, almost none of the works considers a mechanism to prevent unwanted behaviors from an orchestrator and compensate the service's users in case of faulty execution. Such limits hinder the performance of orchestration algorithms. Next, we describe the future research directions given the current status of Mdo algorithms.

V. RESEARCH DIRECTIONS

Based on the review we identify the following research directions for MdO algorithms.

Portability. MdO has not fully matured, as no there is no open standard interface to communicate orchestrators. Related works assume an ad hoc solution to communicate. This lack of standardization hinders the portability of the VNFs. An ideal scenario would be a marketplace that has VNFs that can be instantiated anywhere, anytime. This has the advantages of true elasticity and flexibility, as in case of a problem, they can be migrated easily.

However, ensuring portability brings new challenges, such as state management and support to the services. Managing the VNF's state is challenging because of the availability and privacy constraint. The VNF should remain operational until the re-configuration has finished. Supporting end-to-end service provisioning with legacy networks is challenging since no standard communication interface exists.

Accountability. With the provisioning of more fine-grained network services, providers must establish a chain of responsibility and countermeasures to hold accountable other non-reliable or faulty providers. In the competitive scenario, most works implement a policy to detect and punish these destructive behaviors. However, they fall short of full accountability, as they only implement such mechanisms during the first steps of the lifecycle management of network services.

Accountability in the ideal MdO platform could determine the degree of responsibility in case of failure or not meeting a specific requirement in the SLA. A monitoring system should not only detect local problems but also be able to find problems all along with the end-to-end service. For example, a service failure can be the product of many small failures distributed among different providers; thus, not only it should verify the VNFs independently but also the interaction among them. This transparent method allows providers to prevent conflicts and reduce manual configuration time.

Consistency. MdO cannot rely on global knowledge for conflict resolution. Despite this, few works in the literature consider ensuring consistency for shared services. Most works assume either zero inconsistencies by having a reliable, deterministic network; others presuppose the existence of an internal mechanism that handles such inconsistencies. This lack of proposals to tackle consistency could result from the single-domain origins of NFV, where a central all-knowing orchestrator could take a unilateral decision. However, to ensure functional and non-functional requirements, consistency must be addressed while jointly orchestrating services.

Multiple consistency models can enforce consistency for distributed MdO; however, no single model is appropriate for all uses. Future works in the literature should address different consistency models. Weaker consistency models could resolve some tasks, while others might require full consensus to resolve conflicts.

Privacy. Is a must-requirement for the success of MdO. Most works assume the information being shared does not compromise the privacy and security of participant orchestrators. However, these works provide no guarantees to support this claim; except on one paper that focuses on privacy [6]. With distributed ledgers, encryption has been explored to support privacy in MdO [15]. However, for closed federations, these solutions are overkill because of the latency per operation. Future solutions should include intermediary privacy-aware algorithms such that network service providers trust and share transparent information to support the joint orchestration of services.

Open Federations. Most works in MdO are focused on closed federations. Out of the works we first identified, only two articles were found that address open federations where the number of orchestrators is not known in advance and they can join or leave. Such federations enable a true elastic network service experience as new resources can be offered on the fly, without the disadvantages of a fixed number of providers. However, such flexibility comes with a cost on the complexity of the lifecycle management of network services. Since orchestrators can come and go quickly, no trust is assumed in the federation. Thus, distributed ledgers are mainly now being explored to circumvent problems related to security, privacy, and distributed computation.

We envision smart contracts among network service providers to provide a fully distributed solution for the virtualization of network services. Such contracts fulfill all the tasks of the lifecycle services by writing in a block in the chain. However, the major challenge of such systems is the latency for any operation. While NFV is in the realm of milliseconds; either vote or proof-based consensus algorithms, are in seconds or minutes. For now, they appear only to work on small use cases and scaling seems to be a problem. These incentive mechanisms would allow cooperative works on open federations. Nevertheless, open federations remain an open question in the NFV context.

VI. Conclusion

Multi-domain orchestration is the next evolution for managing the softwarized network services lifecycle. To the best of our knowledge, no survey covered only distributed multi-domain orchestration algorithmic solutions. In this paper, we only considered them and identified key requirements present in cooperative and competitive approaches for multi-domain orchestration. The papers were classified and evaluated based on a proposed taxonomy. We identified some requirements that were fully considered (e.g. interoperability); while others were not (e.g. portability). Some key requirements were absent from cooperative, while others for competitive environments (e.g. secrecy and consistency). Considering such results, we discussed the trade-offs required for a successful orchestration algorithmic solution and some promising research directions for multi-domain orchestration solutions, such as open federations.

Acknowledgment

The research was supported by CONACYT (Grant 708000) and CNRS.

References

[1] Yi, B., Wang, X., Li, K., Das, S. & Huang, M. A comprehensive survey of Network Function Virtualization. *Computer Networks*. **133** pp. 212-262 (2018)

[2] Saraiva de Sousa, N., Lachos Perez, D., Rosa, R., Santos, M. & Esteve Rothenberg, C. Network Service Orchestration: A survey. *Computer Communications*. **142-143** pp. 69-94 (2019)

[3] Guerzoni, R., Vaishnavi, I., Perez Caparros, D., Galis, A., Tusa, F., Monti, P., Sganbelluri, A., Biczók, G., Sonkoly, B., Toka, L., Ramos, A., Melián, J., Dugeon, O., Cugini, F., Martini, B., Iovanna, P., Giuliani, G., Figueiredo, R., Contreras-Murillo, L., Bernardos, C., Santana, C. & Szabo, R. Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: an architectural survey. *Transactions On Emerging Telecommunications Technologies*. **28**, e3103 (2017,4)

[4] Dieye, M., Jaafar, W., Elbiaze, H. & Glitho, R. Market Driven Multidomain Network Service Orchestration in 5G Networks. *IEEE Journal On Selected Areas In Communications*. **38**, 1417-1431 (2020)

[5] Cisneros, J., Yangui, S., Pomares Hernández, S., Pérez Sansalvador, J. & Drira, K. Coordination Algorithm for Migration of Shared VNFs in Federated Environments. *2020 6th IEEE Conference On Network Softwarization (NetSoft)*. pp. 252-256 (2020)

[6] Joshi, K. & Kataoka, K. pSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN. *Computer Networks*. **178** pp. 107295 (2020)

[7] Zhang, Q., Wang, X., Kim, I., Palacharla, P. & Ikeuchi, T. Vertex-centric computation of service function chains in multi-domain networks. *2016 IEEE NetSoft Conference And Workshops (NetSoft)*. pp. 211-218 (2016)

[8] Xu, Z., Zhang, Z., Liang, W., Xia, Q., Rana, O. & Wu, G. QoS-Aware VNF Placement and Service Chaining for IoT Applications in Multi-Tier Mobile Edge Networks. *ACM Trans. Sen. Netw.*. **16** (2020,6)

[9] Kaur, K., Garg, S., Kaddoum, G., Gagnon, F., Kumar, N. & Ahmed, S. An Energy-driven Network Function Virtualization for Multi-domain Software Defined Networks. *IEEE INFOCOM 2019 - IEEE Conference On Computer Communications Workshops (INFOCOM WKSHPS)*. pp. 121-126 (2019)

[10] Quang, P., Bradai, A., Singh, K., Picard, G. & Riggio, R. Single and Multi-Domain Adaptive Allocation Algorithms for VNF Forwarding Graph Embedding. *IEEE Transactions On Network And Service Management*. **16**, 98-112 (2019)
[11] Zhang, C., Wang, X., Zhao, Y., Dong, A., Li, F. & Huang, M. Cost Efficient and Low-Latency Network Service Chain Deployment Across Multiple Domains for SDN. *IEEE Access*. **7** pp. 143454-143470 (2019)
[12] Li, G., Zhou, H., Feng, B., Li, G. & Xu, Q. Horizontal-based orchestration for multi-domain SFC in SDN/NFV-enabled satellite/terrestrial networks. *China Communications*. **15**, 77-91 (2018,5), https://ieeexplore.ieee.org/document/8387988/
[13] Quang, P., Bradai, A., Singh, K. & Hadjadj-Aoul, Y. Multi-domain non-cooperative VNF-FG embedding: A deep reinforcement learning approach. *IEEE INFOCOM 2019 - IEEE Conference On Computer Communications Workshops (INFOCOM WKSHPS)*. pp. 886-891 (2019)
[14] Swapna, A., Rosa, R., Rothenberg, C., Pasquini, R. & Baliosian, J. Policy Controlled Multi-domain cloud-network Slice Orchestration Strategy based on Reinforcement Learning. *2020 IEEE Conference On Network Function Virtualization And Software Defined Networks (NFV-SDN)*. pp. 167-173 (2020,11), https://ieeexplore.ieee.org/document/9289852/
[15] Rathi, V., Chaudhary, V., Rajput, N., Ahuja, B., Jaiswal, A., Gupta, D., Elhoseny, M. & Hammoudeh, M. A Blockchain-Enabled Multi Domain Edge Computing Orchestrator. *IEEE Internet Of Things Magazine*. **3**, 30-36 (2020,6), https://ieeexplore.ieee.org/document/9125428/

Josue Castañeda Cisneros received his M.S. degree in computer science from the INAOE, Puebla. He is currently a Ph.D. candidate at CNRS LAAS in France. His research focuses on Distributed Multi-Domain Orchestration under Network Function Virtualization. His topics of interest include coordination of orchestrators in both open and close federations, respectively.

Sami Yangui is an Associate Professor with INSA, Toulouse. He is member of the CNRS LAAS research team. His research interests include distributed systems and architectures, service-oriented computing and Internet of Things. He is working on different aspects, such as cloud/fog computing, network functions virtualization and content delivery networks. He is involved in different European and International projects, as well as, standardization efforts.

Saul Pomares received the Ph.D. degree in computer science and telecommunications from the INP, Toulouse. Since 1998, he has been researching in the field of distributed systems and partial order algorithms. He is currently a Professor with the computer science department at INAOE, Puebla. He is also an Honorary Researcher with the CNRS LAAS CNRS.

Khalil Drira received the Ph.D. and HDR degrees in computer science from UPS Toulouse in 1992 and 2005. Since 1992, he assumes a full-time research position in CNRS, France. His research interests include cooperative network IoT services, platforms and applications. His research activity addresses topics in this field focusing on Software architectures and communication services. He continues to be involved in national and international conferences and journals, some as an Editor. He serves as a member of the program journals in the fields of software architecture, as well as IoT and Internet networks.