Seamless MANO of Multi-vendor SDN Controllers Across Federated Multi-Domains

Emidio P. Neto[a,b], Felipe S. Dantas Silva[a,b], Lucas M. Schneider[a], Augusto V. Neto[b,c] and Roger Immich[b]

[a] *LaTARC Research Lab – Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), Natal/RN, Brazil*
[b] *Federal University of Rio Grande do Norte (UFRN), Natal/RN, Brazil*
[c] *Instituto de Telecomunicações, Aveiro, Portugal*

ARTICLE INFO

Keywords:
Software-defined Networking
Orchestration
MANO
Agnostic

ABSTRACT

The upcoming 5G networks promise to provide end-to-end network service delivery that spans across Cloud-Network federated multi-domains while providing monolithic service guarantees. By adopting a Cloud-Network federated multi-domain approach, heterogeneity raises a number of challenging Management and Orchestration (MANO) perspectives, especially concerning the need to deal with SDN controllers of multi-vendor approaches that lack a standard Northbound API. In this work, we tackle the issue of providing a seamless MANO of SDN Controllers running in Cloud-Network federated multi-domains. Owing to the weaknesses and limitations of the related works, we propose the ***W****AN* ***I****nfrastructure* ***M****anager* ***A****gnostic* (**WIMA**) that enables a seamless and vendor-agnostic MANO abstraction to run on top of federated SDN multi-domains. The WIMA provides a common Northbound API for external triggering, maintains a global topology view of the federation as a whole, and deals with each SDN Controller directly by deploying an ontology-based scheme for efficient Northbound API mapping. The effectiveness and performance impact of WIMA was assessed in an emulated testbed with homogeneous ("Hom") and heterogeneous ("Het") multi-domain SDN control-planes, along with a varying density of active Tenants which simultaneously makes flow stress data connections during an experimental time of 900 seconds. The obtained results reveal that WIMA's MANO abstraction system is able to connect around 52.66% ("Het") and 86.87% ("Hom") more end-to-end data flows across the federated SDN multi-domains while adding greater agility 52.72% ("Het") and 85.27% ("Hom") than the rival Baseline solution. Thus, the WIMA's central logic has proven to be a suitable and feasible means of ensuring the MANO framework's efficiency atop the multi-domain SDN Controllers within a Cloud-Network federation while optimizing theas well as significantly lower operation time.

1. Introduction

The upcoming 5G network system promises to make rapid progress in supporting innovative mobile communications for Ultra-Reliable and Low Latency Communication (URLLC), massive Machine Type Communication (mMTC), and enhanced Mobile Broadband (eMBB) use cases [1, 2]. The 5G architecture follows a disruptive Software-Defined Networking (SDN) [3] native design, which enables application-controlled programmability and network resource manageability in a dynamic, granular, and scalable fraction. The list of evolutionary 5G network innovative features includes monolithic service delivery within isolated and quality-guaranteed network infrastructures (e.g., network slices [4]), which can span across multiple network operators inside a federated cloud [5], thus enabling a Cloud-Network federation system.

Cloud-Network federation gained momentum with the advent of 5G networks, in which multiple network carrier domains seek to provide communication services among different cloud-federated providers [6]. Cloud-Network federation refers to a set of different cloud and network provider domains, each formally disconnected and forming distinct internal structures. These collectively adhere to share resources to provide customers with increased benefits while reducing the total cost of ownership [7]. Figure 1 provides an example of a Network-Cloud federated multi-domain scenario.

As can be seen from Figure 1, Cloud-Network federated providers operate in domains that are geographically dispersed and owned by independent organizations committed to complying with a joint authority working within a multi-cloud Management and Orchestration (MANO) framework. Hence, a Cloud-Network federation can be seen as a single structure with an expanded domain of resources that allows it to provide a service (through slicing technique, for instance) [8]. The Novel Enablers for Cloud Slicing (NECOS) project [9] exemplifies this conceptual trend, by enabling Cloud-Network federated domains to offer end-to-end multi-tenant service delivery over high-level isolated and manageable cloud-network slice instances. Our research tackles the problem of enabling the common MANO framework to seamlessly orchestrate multi-domain SDN control-planes to accomplish the complex task of end-to-end provisioning networking across the Cloud-Network federation.

---

*Corresponding author at: Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), Brazil.

ORCID(s):

[1] *Email addresses*: emidio.paiva@ifrn.edu.br (E. Neto), felipe.dantas@ifrn.edu.br (F. Dantas), lucas.schneider@ifrn.edu.br (L. Schneider), augusto@dimap.ufrn.br (A. Neto), roger@imd.ufrn.br (R. Immich).

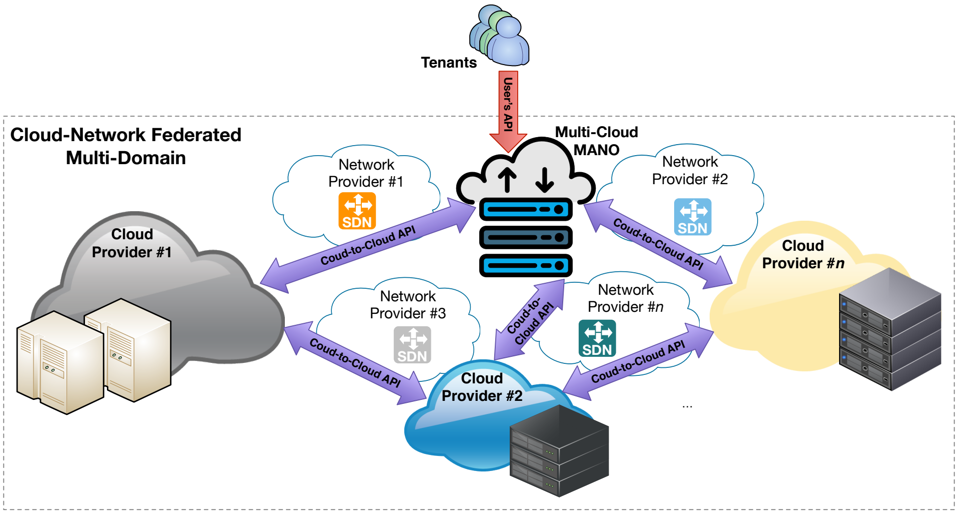


Figure 1: Illustration of a Cloud-Network federated multi-domain scenario

Each network operator tends to follow a single unified SDN control plane for the MANO of network services and resources in the underlying domain [10], as Figure 1 illustrates. When it joins a Cloud-Network multi-domain federation, each network operator is naturally accompanied by its ecosystem capacity and enabling technologies, including its SDN controller. Each vendor implements SDN Controllers by adopting specific approaches. For instance, Ryu [11], ONOS [12], and OpenDaylight [13] stand out as leading solutions for OpenFlow [14] domains, which are differentiated from each other in terms of interfaces, semantics, data structure, functionalities, and other critical factors [15]. Apart from this, it is well-known that operators are very reluctant to change their ecosystem technologies aggressively (in this paper, the SDN control plane) to join the federation for a number of significant reasons. Hence, the task of accelerating the deployment of end-to-end connectivity atop a Cloud-Network multi-domain federation is very challenging. The reason is that heterogeneous SDN controller solutions must coexist, interwork, and share resources and information with each other, which adds to the complexity of MANO operations when adopting a fully cooperative and abstracted approach [9].

Introducing a MANO solution to exclusively handles each SDN domain inside the federation is high costly, and it increases the complexity of responding to the heterogeneity discussed earlier. Moreover, providing information and knowledge about the current networking topology and conditions, in terms of both available resources and controller-specific capabilities in a global setting, raises a further related challenge. We believe that when the MANO of multi-vendor SDN Controllers incorporates different federated Cloud-Network multi-domains, it must be carried out seamlessly by a single abstraction. However, the lack of a standard Northbound Application Programming Interface (API) has proved to be the main problem of obtaining a seamless MANO abstraction [16]. Furthermore, this kind of MANO abstraction must build and maintain a global knowledge-based structural topology to enable holistic computing and decision-making with regard to the SDN as a whole. Thus, the MANO abstraction will pave the way for an end-to-end and in-depth knowledge of the current conditions in the entire federated Cloud-Network multi-domain infrastructure.

In this work, we tackle the problem of providing seamless MANO that can allow multi-vendor SDN Controllers to establish end-to-end network service delivery that spans the federated Cloud-Network multi-domains. The proposed solution, called **W**AN **I**nfrastructure **M**anager **A**gnostic (WIMA), adds a straightforward common MANO framework on top of the multi-vendor SDN Controllers that include the different federated Cloud-Network multi-domains. WIMA is a promising strategy for managing network systems across a wide range of unrelated network and cloud domains in an integrated, automated, and seamless fashion. The seamless MANO abstraction that WIMA holds is invoked through a common Northbound API in a modular and vendor-independent scheme.

The main research contributions made by this work are threefold. First, we have investigated existing solutions by examining the MANO of heterogeneous SDN Controllers uniformly. Second, we proposed the use of WIMA, which is designed with MANO functions that can operate seamlessly for end-to-end QoS guaranteed connectivity across multiple (and either homogeneously- or heterogeneously-structured) domains inside a Cloud-Network federation. Finally, we have designed a vendor-agnostic Northbound API to allow that external management applications can trigger WIMA's seamless MANO functions in a standard way for enforcing multi-domain SDN Controllers in their specific Northbound API. A series set of experiments were conducted through a testbed approach to assess the effectiveness and performance of the WIMA scheme. The performance of WIMA was assessed concerning the most significant related work (designated as the Baseline); this took the

E2E Flow Setup Time and E2E Flow Activation impact on the testbed as key Performance Indicators (KPIs). The experimental outcomes suggest that WIMA outperformed the Baseline agent-based rival solution through its ability to provide enhanced performance rates for the E2E Flow Setup Time (52.72% in "Het" and 85.27% in "Hom") and E2E Flow Activation (52.66% in "Het" and 86.87% in "Hom") KPIs, for the Cloud-Network multi-domain federation homogeneous/heterogeneous testbed settings.

The remainder of this work is structured as follows. Section 2 summarizes the related literature in the area. This is followed by Section 3, which outlines the proposed WIMA system. The implementation and assessments are examined in Section 4, along with the discussion of the results. Finally, Section 5 wraps-up the paper with concluding remarks and recommendations for future work.

2. Related Work

Our survey on how research endeavors are consolidated in the literature revealed several SDN Controller orchestration solutions designed for provisioning a means to visualize underlying network devices graphically. The authors of [17, 18] introduce solutions that are tailored to display the layout of the in-topology devices. In [19], the authors put forward OpenGUFI, a system that provides a Graphical User Interface (GUI) for network topology abstraction, which only allows the operator to visualize the connections between mobile clients and wireless access points, along with the respective flowchart rules. Although they support the need for a knowledge of topology maintenance, all these solutions lack control-plane functions that can allow the operator to manage and control the underlying in-topology nodes, which means they are unsuitable for seamless MANO.

Another set of previous studies follows an architectural design that relies on third-party technologies to obtain network operator license acquisitions for activation. The RUNOS controller [20] is an example of a device that is based on the Maple Computation algebra software [21], which is used to optimize applications. Although RUNOS avoids system overhead by managing processes through multi-threading, it relies on proprietary protocols and non-standard Northbound APIs, which increases the complexity of the system and requires the network operator to have specialized knowledge of third-party technologies. Finally, RUNOS requires individual triggering for each of the different controllers, which contradicts the concept of seamless MANO.

In a recent study by [22], the authors made out a case for allocating a Wide-area Infrastructure Manager (WIM) instance in an on-demand manner to support softwarized network slicing. The main goal of WIM on-demand is to manage connectivity attributes prescribed for full end-to-end network slices in a flexible and adaptable way. In this work, the authors show how to allocate a different WIM instance in an on-demand, each dealing with a particular network slice instance rather than having one for the whole network. Although they allow different SDN Controllers (WIMs) to coexist in multi-domains, the WIMs run as independent instances. Thus, the network operator has to configure the WIMs in advance following the targeting network slice and use multiple Northbound APIs for MANO. This strategy makes it a complex task to interoperate with multi-vendor SDN Controllers, since applications that have been designed for a particular SDN Controller, cannot be used by any other. Onix [23], (based on the NOX controller [24]), IRIS-CoMan [25] (based on the Floodlight controller), Kandoo [26] (based on Kandoo controller), DISCO [27] (based on the Floodlight controller), and HyperFlow [28] (available as a NOX driver application are examples of these kinds of solutions).

For our research purposes, the OrchFlow [29] stands out as the most significant of all the previous related works since it supports the MANO of multi-vendor SDN Controller systems that coexist inside domains. The OrchFlow architecture has an agent-based design, which provides a hierarchical workflow to carry out MANO functions in each intended domain of the underlying network infrastructure. The OrchFlow's agent-based approach imposes an additional abstraction layer for each domain, which is meant to interact with the high-level abstraction in a proprietary way, and then invoke each SDN Controller accordingly. Furthermore, OrchFlow relies on the third-party Neo4J graph-oriented database for managing topology state and provisioning interfaces for external access. As a result, dependency on third-party technology makes it challenging to implement any additional features for OrchFlow.

Although the schemes listed in this section offer particular benefits, some of them cannot be coordinated with the leading SDN Controllers in the market. Moreover, a few of them only offer the graphical visualization features of the underlying topology and lack the support of any SDN control functions. Another set of schemes is dependent on third-party technologies, which are expensive, rely on proprietary protocols, and deploy non-standard Northbound APIs, thus increasing the system's complexity. Our findings enabled us to compile a list of the minimum capabilities we believe are needed to ensure a MANO solution with a capacity for end-to-end network service delivery over federated multi-domains entailing multi-vendor SDN Controllers:

- **R1:** provides knowledge and information about a global topology view, including that of all the federated domains and respective SDN Controller systems;
- **R2:** enables mechanisms to compute QoS-guaranteed end-to-end paths across the federated domains;
- **R3:** designs an architecture-independent on third-party technologies;
- **R4:** offers a standard Northbound API for external applications to request an end-to-end network service application setup;

- **R5:** provisions mechanisms tailored to enable seamless MANO functions on top of multi-vendor SDN Controllers.

Table 1 makes a comparison between the related work discussed in this section and the WIMA proposal, with regard to the list of requirements outlined above, to distinguish those that provide seamless MANO support over federated multi-domains that incorporate multi-vendor SDN Controllers.

Table 1: Comparison between the related works and the proposed WIMA system

Proposal	**Year**	**R1**	**R2**	**R3**	**R4**	**R5**
HyperFlow [28]	2010	✓	✗	✗	✗	✗
Onix [23]	2010	✓	✓	✗	✗	✗
Kandoo et al. [26]	2012	✓	✗	✗	✓	✗
DISCO [27]	2013	✓	✓	✗	✗	✗
Huang et al. [17]	2014	✓	✗	✗	✗	✗
Mantoo [18]	2015	✓	✗	✗	✓	✗
OpenGUFI [19]	2015	✓	✗	✗	✗	✗
IRIS-CoMan [25]	2015	✓	✗	✗	✓	✗
RUNOS [20]	2015	✓	✗	✗	✗	✗
OrchFlow [29]	2018	✓	✓	✗	✓	✗
WIM on demand [22]	2020	✓	✗	✗	✗	✗
WIMA	**2020**	✓	✓	✓	✓	✓

It should be noted that the OrchFlow [29] is the only related work solution that supports the MANO of multi-vendor SDN Controllers. However, the agent-based architecture design that OrchFlow follows depends on third-party technologies to orchestrate the high-level and domain-level abstractions, which then trigger each agent inside the intended domain accordingly. A thorough review of the literature was conducted, and it can be confirmed that, to the best of our knowledge, there is no other available solution capable of providing the network operator with the minimum capabilities. In light of these limitations, we are able to recommend suggest WIMA since there is a need for a new solution with capabilities of seamless MANO end-to-end communications across Cloud-Network federated multi-domains.

3. Towards WIMA Design

The architecture, concepts, and principles of WIMA are designed to address the complex and challenging task of provisioning end-to-end QoS-guaranteed network service delivery across Cloud-Network federated multi-domains (as discussed in Section 1). To settle these issues, WIMA adopts a straightforward approach that enables it to carry out MANO features seamlessly over either homogeneous (a single SDN Controller approach) or heterogeneous (featuring multi-vendor SDN Controllers) domain structure inside the federation. As a means of achieving this goal, WIMA adds a single holistic MANO abstraction framework layer that runs on top of the Cloud-Network federated multi-domains and can be accessed through a common Northbound API. By contrast with the multi-agent-based OrchFlow rival solution, WIMA triggers each SDN Controller directly by following an ontology-based mapping strategy, without needing the assistance of any intermediate solutions (such as in-domain running agents). Furthermore, the role of WIMA entails supporting a set of SDN Controller specific features, such as the following: (i) maintaining a consistent global view of the entire SDN topology of the Cloud-Network federation; (ii) a feature to (re)orchestrate virtual network elements at the run time to commit end-to-end (intra- and inter-domain) QoS-guaranteed SDN path computing; (iii) interfacing with SDN Controllers in their specific technologies; (iv) quickly responding to asynchronous network events that occur in all federated multi-domains; just to name the main features. As a result, the SDN Controllers of each Cloud-Network federated domains become lightweight by reacting to incoming requests by WIMA to carry out network state setup, statistical delivery, alarm system, and other features that the local Southbound API supports.

3.1. The design of WIMA Architectural Components

The design of the WIMA follows a modular architecture consisting of fundamental components that interwork to commit end-to-end network service delivery across the SDN-enabled domains within the Cloud-Network federation. Figure 2 shows the modular architecture of the WIMA proposal, and highlights its relationship with the Management and the Cloud-Network Federation planes.

The Management Plane refers to external services and applications running atop the network (named in Figure 2 by Management Applications), which are designed to trigger WIMA through a common, well-defined, and vendor-agnostic interface, designated as a Management Plane Northbound API (MPNB-API). The definition of MPNB-API is based on the RESTful technology, in which external management applications can request WIMA to carry out MANO functions seamlessly in the underlying Cloud-Network federated multi-domains. The WIMA-supported MANO functions will run in a fully abstracted manner, regardless of whether each in-domain is running SDN Controller Northbound API specifics (i.e., seamlessly). The list of MPNB-API supported operations and protocol specifications is outlined below.

Global-view Topology: this consists of providing real-time knowledge about the entire topology that configures the Cloud-Network federation. WIMA books the topology state knowledge in a graph-compatible format, including in-depth information about all the federated SDN domains subject to seamless MANO.

E2E Flow Setup: devoted to provisioning an end-to-end QoS-guaranteed SDN path so that two or more destination hosts located inside the same domain can be interconnected in different domains of the Cloud-Network federation, or even in a domain outside the federation. To achieve this, the MPNB-API receives both source and destination addresses and calculates the best SDN path for connecting the two

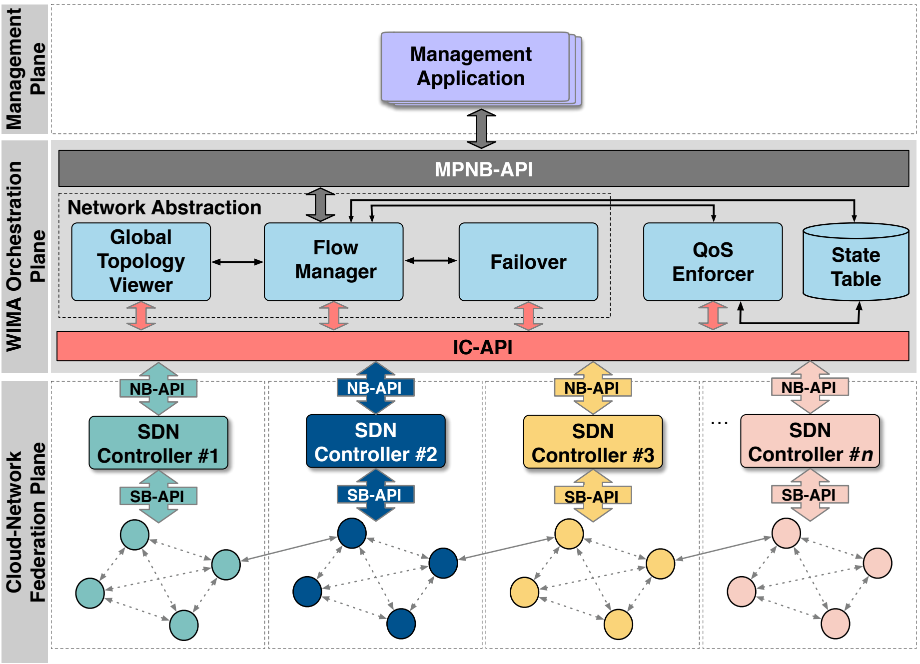


Figure 2: Modular Architecture

hosts (or the egress router of the federation, in case there is a host outside the federation). WIMA leverages the IC-API to enforce flow table modifications, add measurements, and packets control rate settings in the nodes along the computed SDN path.

The Inter-Controller API (IC-API), in turn, plays the critical role of establishing a single point of communication between the SDN Controller nodes running in the underlying orchestrated multi-domains of Cloud-Network federation, in its specific Northbound API technology, and the other components of the WIMA architecture. The IC-API is characterized by its ability to access different domain-federated SDN Controllers in a monolithic way, without the assistance of any intermediate solutions (contrary to OrchFlow, which needs specific agents running atop each SDN Controller, inside the respective domain). The IC-API design adopts an ontology-based approach to represent the knowledge needed for mapping different Northbound APIs to enforce an SDN Controller to setup the intended network state, or gathering statistical data, in the multi-domains. With this goal in mind, on inferring information from the knowledge map, the IC-API composes a matching message, fills all the respective fields with data derived from the previous message, and sends the translated message to the targeted SDN Controller.

The IC-API formally represents knowledge maps with detailed information about the different domain-federated SDN Controllers, including the API specification, return data format, message parameters, authentication methods, controller version, and other features. The ontology reasoning is done by the IC-API feature-matching algorithm, which is designed with a sequence of actions to map one Northbound APIs into another one that a targeted SDN Controller supports. In the general case, the IC-API feature-matching algorithm specifies correspondences between Northbound APIs, meaning the translation of functions that the incoming Northbound API message indicates (used by an external management application to trigger WIMA in a common way) carries into an outgoing Northbound API message (whereby WIMA uses to invoke a domain's SDN Controller). The IC-API feature-matching algorithm must be lightweight so that it can allow quick ontology reasoning for a scalable computing approach. In light of this, we designed the ontology map using OWL (Ontology Web Language) semantics. We speeded up the OWL reasoning while improving scalability by implementing the translation knowledge through a hash map representation. Hence, the WIMA architecture is ready to process simultaneous active tenants by achieving low response times. Aside from this, the IC-API ontology reasoning algorithm benefits from an absence of incoherence since the knowledge map includes low granularity for the SDN Controller approaches, along with a low degree of API similarity.

The semantic of each request message directed to the IC-API consists of 2 fields: the first is responsible for identifying the type of request message, which can be classified as (*i*) Topology, (*ii*) QoS, and (*iii*) Flow Entry. The second field identifies the SDN controller(s) involved in the operation. During the translation of a request message to communicate with different SDN controllers, the IC-API employs the feature-matching algorithm by retrieving the request type identifier and the knowledge map controller identifier (SDN Controller Type #N API Mapping). Thus, it is possible to infer the method from the knowledge map and other parameters related to the Northbound API of the targeted in-domain SDN controller. Finally, the IC-API delivers the request message translated into the syntax that the targeting SDN controller implements.

It should be noted that, it is necessary to consult the respective API mappings to enable the IC-API to carry out the feature-matching for different SDN controllers. Thus, in federated multi-domains entailing heterogeneous SDN control-planes, the number of request message translations will increase exponentially with the number of domains selected during the E2E Flow Setup. However, if the multi-domains are homogeneous in terms of SDN controllers, the translation can only be done once in a straightforward manner, whereas translated messages are replicated (properly filled) to each destination domain. This approach is significantly more cost-effective in terms of computing than with multi-domains scenarios of heterogeneous SDN control-planes. Finally, but not least, another advantage of the ontology we designed for integration lies in its external data representation through YAML syntax configuration files. The mappings and translations based on technologies' vocabulary are performed automatically after these YAML files have been read, which allows updated parameters and methods of new SDN Controllers to be added without the need to recompile the IC-API code. This provided the IC-API with an efficient and highly-scalable moldable approach.

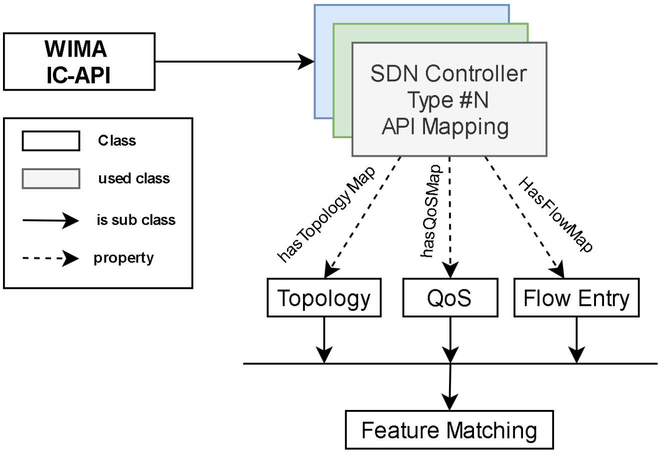


Figure 3: WIMA SDN Controllers ontology map of knowledge

Two blocks, namely the Network Abstraction and the QoS Enforcer, along with one state table, represent the fundamental building blocks that form the WIMA modular architecture:

3.1.1. Network Abstraction

The Network Abstraction is the WIMA Architectural component responsible for facilitating the complex task of providing monolithic services and orchestrating multi-vendor SDN Controllers within Cloud-Network federated multi-domains. In achieving this, the Network Abstraction undertakes critical SDN service MANO tasks in a single system (e.g., global SDN path computing, flow table setup, and resilience). On the other hand, the in-domain SDN Controllers remain lightweight, by harnessing the local Southbound API (SB-API in Figure 2) for the purpose of enforcing resource state in the underlying domain nodes, in response to an incoming Network Abstraction request, as well as warning WIMA about the detection of asynchronous events (e.g., link breaks). Furthermore, it also provides a communication interface to ensure easy access to the SDN Controllers' functionalities. The Network Abstraction comprises three subcomponents, namely the Global Topology Viewer, the Flow Manager, and the Failover. These subcomponents follow a flexible approach to cope with each network operator's particular needs, as listed below.

- Flow Manager: stands to the entry point logic of the Network Abstraction component inside the WIMA Architecture. The Flow Manager primarily tackles optimal SDN path computation for incoming network flows. This component is aware of the entire network infrastructure of the underlying Cloud-Network federation, which spans the multiple federated domains.

- Global Topology Viewer: provides a global (holistic and ubiquitous) view of the entire Cloud-Network federated multi-domain infrastructure under the control of the WIMA MANO common framework. The topology knowledge's global view maintains UpToDate attributes about all the SDN-enabled switches, interconnection links, resources, and respective SDN Controllers of the multi-domains. The Global Topology Viewer communicates with each SDN Controller to obtain knowledge about the underlying SDN topology, along with the current capacities of in-domain nodes. It is also responsible for receiving asynchronous network events.

- Failover: provides resilience capabilities by seamlessly and automatically responding to asynchronous network events that the SDN Controllers of the underlying Cloud-Network federated multi-domains deliver (assisted by the Global Topology Viewer). For instance, the Failover component seeks to quickly reroute data flows affected by a link failure event, in an attempt to avoid service disruptions while keeping connectivity at its best over time.

3.1.2. QoS Enforcer

The QoS Enforcer architectural WIMA's component is designed to trigger different SDN Controllers, in their specific Northbound API, so that they can allocate resources for provisioning an end-to-end QoS-guaranteed networking service. The QoS specifications that the network operator provides in the service description, are used by the QoS Enforcer to configure bandwidth reservations, traffic conditioning, queue priority rules, and other QoS features. For instance, the QoS Enforcer component orchestrates both *meter tables* and *queue* features at selected on-path nodes within the SDN domains by deploying the OpenFlow Southbound API.

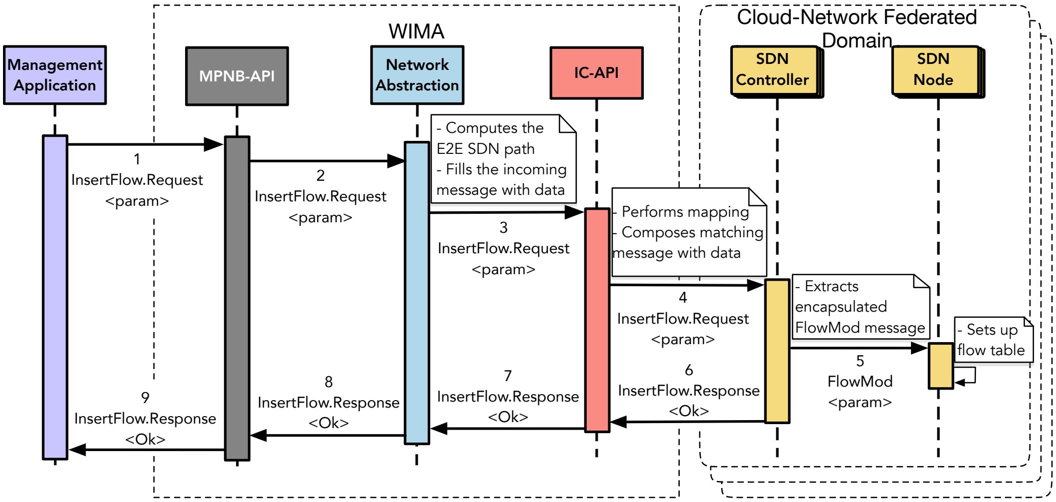


Figure 4: Sequence diagram of new flow setup procedures

3.2. WIMA Use Case

This section describes how WIMA operates in a Cloud-Network federated multi-domain infrastructure for seamless MANO by providing end-to-end communication service delivery. The use case assumes WIMA can allow a network operator management application by requesting the lifecycle of a new incoming flow (i.e., the flow creation and QoS queue setup workflows) service along with a list of application-specific requirements. Figure 4 illustrates the sequence of workflows of the E2E Flow Setup operation and messages exchanges between the external management application, the WIMA interworking components, and the SDN infrastructure.

Figure 4 shows a network operator on-premises management application leveraging the MPNB-API to request the installation of a new flow (i.e., E2E Flow Setup operation), provided by the `InsertFlow.Request` message using customized parameters (Step 1). On receiving the `InsertFlow.Request` message (Step 2), the Network Abstraction, derives the destination network and data-plane QoS requirements by means of the flow Manager subcomponent. Afterward, the best end-to-end SDN path computing is performed, relying on the assistance of the global view topology obtained through the Global Topology Viewer subcomponent (Step 3). Following this, it invokes the IC-API by informing the in-domain SDN path, along with the SDN Controller type.

In order to facilitate understanding the feature-matching algorithm, consider a selected SDN domain under ONOS control to handle the E2E Flow Setup operations. By following the SDN path and ONOS' type representation, that the Network Abstraction component computes (Step 3), the IC-API runs the feature-matching algorithm to translate the intended message syntax. In the present use case, the feature-matching algorithm returns the command correspondences of the ONOS controller Northbound API 'flows' endpoint (in Ryu is the API 'flowentry' endpoint and 'config' for the OpenDaylight). Following this, the IC-API forms the new FlowMod matching message that has been properly filled, encapsulates it into an `InsertFlow.Request`, and delivers it to the destination domain (Step 4) so that the SDN Controller can process it accordingly. This strategy allows customized versions of controllers to be mapped and used within the WIMA ecosystem.

On the SDN domain side, the SDN Controller extracts from the incoming `InsertFlow.Request` message all the in-domain SDN nodes that form the SDN path (computed by the Network Abstraction component). Assuming it is an OpenFlow-enabled domain, the SDN Controller sends a `FlowMod` message (Step 5) towards each of the on-Path OpenFlow switches, which will execute the modified version of the respective state table accordingly. On completing the in-domain SDN path creation, the SDN Controller composes an `InsertFlow.Response<Ok>` message and sends it back to WIMA (Step 6) to confirm that the corresponding request procedure has been accomplished. Finally, WIMA propagates the `InsertFlow.Response<Ok>` message internally until it reaching the requesting Management Application (Steps 7-9), and thus shows that the E2E Flow Setup operation has been completed.

4. Evaluation of Results and Contributions

This section provides a set of procedures to assess the effectiveness and performance of the WIMA proposal, a MANO framework designed to deal with multi-vendor SDN-controllers running on Cloud-Network federated multi-domains seamlessly. WIMA provides a standard vendor-agnostic Northbound API so that external management applications can request end-to-end network service delivery. With the goal of achieving accurate benchmarking, we implemented the whole WIMA architecture (out-

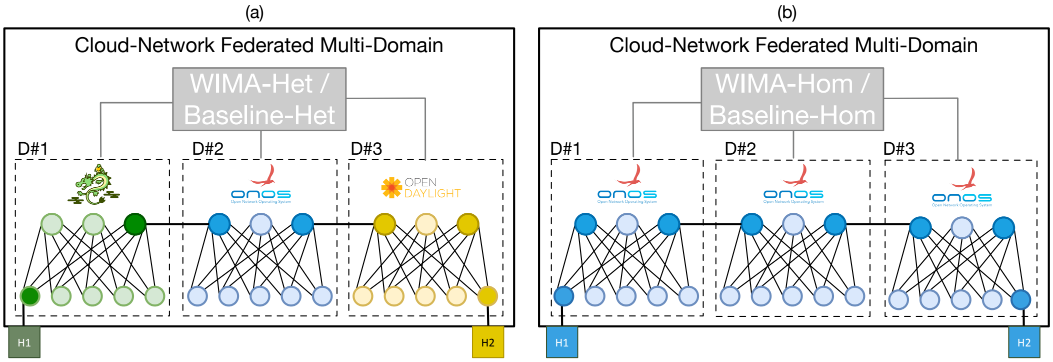


Figure 5: Layout of the Cloud-Network federated multi-domain infrastructures featuring (a) WIMA-Het and Baseline-Het, (b) WIMA-Hom and Baseline-Hom testbed deployments

lined in Section 3.1) and an OrchFlow-alike solution (based on the definitions that [29] provides), which is designated as Baseline. The choice of OrchFlow for benchmarking is justified by the fact that it meets the most significant number of requirements (see Table 1) with regard to the other related works.Incidentally, there was no change to the OrchFlow solution in terms of logic, fundamentals, and key concepts.

We designed a network topology to represent a federation that entails three (3) interconnected OpenFlow-enabled SDN domains (namely *D#1*, *D#2*, and *D#3*). The whole topology features a homogeneous broadband and wired networking capacity of 1,000 Mb/s. A domain connects its right-hand spine switch to the left-hand spine switch of the immediate next domain. The *H1* node is connected to the Cloud-Network federated domain *D#1*, and *H2* node to the Cloud-Network federated domain *D#3*. The whole federation infrastructure is logically virtualized into an Intel Xeon Silver 4114 CPU 2.20GHz (16 vCPUs), 32GB RAM, and Ubuntu Server (18.04) 64-Bit operating system, which harnesses the features of the Mininet[2] emulator version 2.2.2. Two (2) trials have been carried out to determine the impact that both WIMA and Baseline capabilities have on different Cloud-Network federated scenarios. These have the following layout: (*i*) On the one hand, the Heterogeneous testbed configuration deploys a full heterogeneous SDN control-plane, in which each domain is governed by a different SDN Controller vendor approach, namely Ryu (*D#1*), ONOS (*D#2*), and OpenDaylight (*D#3*); (*ii*) the homogeneous testbed configuration, on the other hand, involves a homogeneous environment where all the Cloud-Network federated domains are locally subjected to the ONOS SDN Control. Figure 5 sketches the layout of the network topologies we adopt in the experiments, which depicts the testbed settings (a) for the WIMA-Het and Baseline-Het experiments, and (b) for the WIMA-Hom and Baseline-Hom trials.

The evaluation experiments are mainly carried out(*i*) to validate the WIMA architecture and determine whether it supports the list of features that are listed in Section 3, along with (*ii*) to assess the effectiveness and performance of the MANO functions of WIMA by seamlessly handling multi-domain SDN control-planes. The WIMA architecture is validated by determining the effectiveness of establishing a set of end-to-end network flows that depart from node *H1* and move towards node *H2*, during the course of the experiment. With regard to the performance assessment of WIMA, the evaluation methodology employed follows the RFC 8456 [30] guidelines and recommendations. Our analysis focuses on the abstraction overhead of both WIMA and the Baseline solutions by taking as Key Performance Indicator (KPIs), the total amount of time to setup end-to-end data flows (E2E Flow Setup Time) along with the flow activation rates in a per-second granularity (E2E Flow Activate). Both WIMA and the Baseline solutions are run in testbed sets with a homogeneous and heterogeneous control-plane, where the outcomes are compared and analyzed.

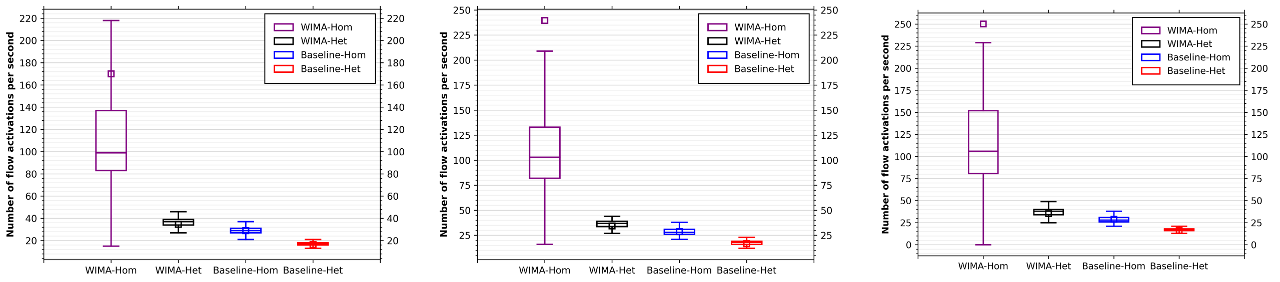
We confirmed the consistency of the outcomes and achieved a confident interval of 95% by carrying out 100 trials of the same experiment. A proof of concept was realized by keeping track of the whole lifecycle that both WIMA (depicted in Figure 4) and Baseline solutions accomplish to set up a new flow in the “Het” and “Hom” testbed configurations. The solutions are designed to operate in accordance with the workflow set out below:

- WIMA-Hom and WIMA-Het set of experiments: (1) Tenant to WIMA; (2) WIMA to SDN Controllers; (3) SDN Controllers to Network Devices; (4) SDN Controllers to WIMA; (5) WIMA to Tenant; and (6) *H1* to *H2*.
- Baseline-Hom and Baseline-Het set of experiments: (1) Tenant to OrchFlow-like abstraction; (2) OrchFlow-like abstraction to OrchFlow-like agent; (3) OrchFlow-like agent to SDN Controllers; (4) SDN Controllers to SDN Switches; (5) SDN Controllers to OrchFlow-like agent; (6) OrchFlow-like agent to OrchFlow-like abstraction; (7) OrchFlow-like abstraction to Tenant; and (8) *H1* to *H2*.

The settings and test characteristics for the evaluation of the SDN-supported cloud federation environment follow pa-

---

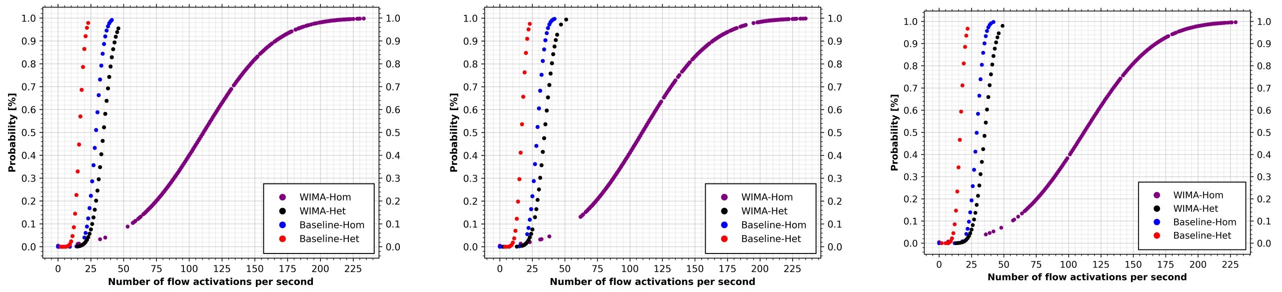
[2] http://mininet.org



(a) 50 active Tenants

(b) 100 active Tenants

(c) 150 active Tenants



(d) 50 active Tenants

(e) 100 active Tenants

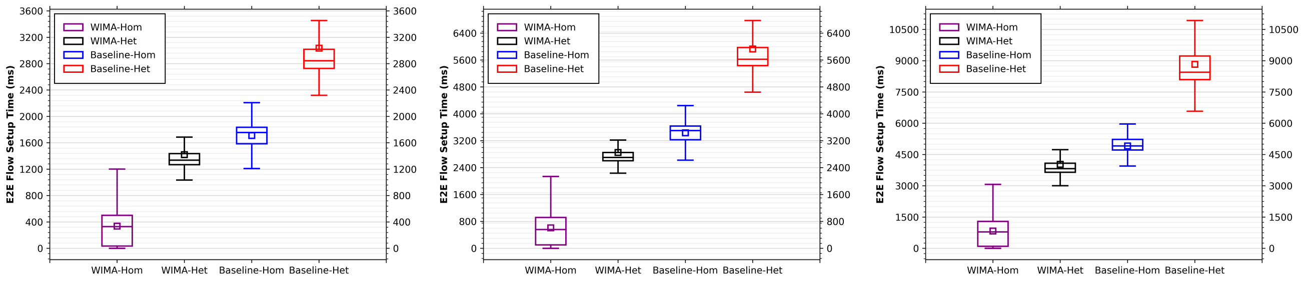
(f) 150 active Tenants

Figure 6: Variability and Cumulative Distribution Function (CDF) of flow activation per second that WIMA-Hom, WIMA-Het, Baseline-Hom, and Baseline-Het solutions have on a varying granularity of Tenants that generate concurrent flow connection requests throughout the experimental time.

rameters and deployments similar to those in [31]. A stress test was performed in the emulated testbed before the experimental runs. Our findings revealed that a granularity of above 150 simultaneous tenants affects several failure exceptions. In this case, both WIMA and the SDN controllers cannot operate in the testbed, leading to interruptions in the experiment. In light of this, the set of experiments comprises three (3) large-scale testbeds with a granularity of 50, 100, and 150 simultaneous Tenants running concurrent connections testings throughout the experimental time that lasts 900 seconds (i.e., 15 minutes). The WRK HTTP benchmarking tool was adopted to assess the solutions in elastic conditions similar to those of a real-world scenario, in contrast with fully controlled simulation experiments that normally adopt a static load limit and provide a predictable evaluation. The WRK tool aims at keeping the number of per second setup flow requests as high as possible until the end of the experimental period. The purpose of WRK benchmarking is to stress a solution by achieving higher activated flow rates, along with the time spent on handling requests, in a per-second granularity during the experiment. Regarding benchmarking, the WRK tool provides the following list of key statistics: total number of offered requests, total data transfer, average latency, and standard deviation rates of connection requests accomplished. Figure 6 shows the statistics for the activated flow rates and respective cumulative distribution, to assess the effectiveness of both WIMA and the Baseline solutions in the "Hom" and "Het" testbed configurations, with a varying number of simultaneous Tenant-offered requests.

The numerical outcomes show that the performance of WIMA-Hom and WIMA-Het to activate flows in a per-second granularity are 170.05 and 34.62 (50 Tenants, Figure 6a), 239.56 and 34.41 (100 Tenants, Figure 6b), and 250.16 and 35.27 (150 Tenants, Figure 6c) respectively. On the other hand, the Baseline solution was able to activate the following for the "Hom" and "Het" testbed configurations: 28.88 and 16.39 flows per second when 50 active Tenants are running; 28.72 and 16.72 flows per second when 100 Tenants are active; and 29.03 and 16.27 in the tests with 150 active Tenants. Hence, the outcomes suggest that WIMA's seamless MANO approach outperforms the Baseline rival in all the sets of the experiments, namely 83.01% (50 Tenants), 88% (100 tenants), and 88.39% (150 Tenants) activated data flows per second for the "Hom" trials. In terms of the "Het" trials, WIMA achieves an enhanced performance of 52.67% (50 Tenants), 51.41% (Tenants), and 53.85% (150 Tenants) with regard to the Baseline solution. This means that the average outperforming rates of WIMA over the Baseline rival solution are 86.47% in the "Hom" testbed configuration and 52.67% in the "Het" experiments.

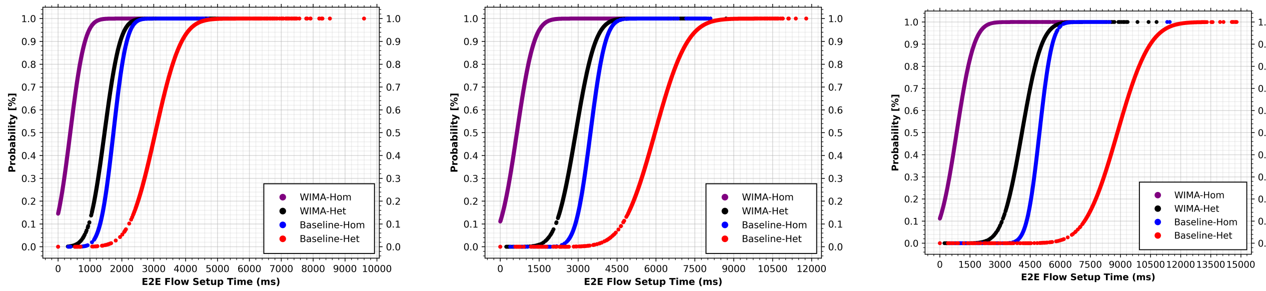
The Cumulative Distribution Function (CDF) statistics shown in Figures 6d, 6e, and 6f , are based on the total number of requests/second and time response/second, and presents WIMA solution with a higher probability of setting up more flows per second than the rival Baseline solution, throughout the experiments. For instance, the probability of WIMA accomplishing more than 70 flow requests



(a) 50 simultaneous flow requests

(b) 100 simultaneous flow requests

(c) 150 simultaneous flow requests



(d) 50 simultaneous flow requests

(e) 100 simultaneous flow requests

(f) 150 simultaneous flow requests

Figure 7: Variability and CDF of the Setup Time that influence the WIMA-Hom, WIMA-Het, Baseline-Hom, and Baseline-Het solutions when responding to 50, 100, and 150 simultaneous E2E Flow Setup requests during the experimental period.

per second is 82.99% (50 Tenants), 82.45% (100 Tenants), and 83.32% (150 Tenants) in the “Hom” trials, whereas it is 75.41% (50 Tenants), 74.9% (100 Tenants), and 78.58% (150 Tenants) for more than 30 flow requests per second during the “Het” trials. On the other hand, the Baseline solution has a probability of a) 58.83%, 60.54%, and 58.34% of being able to connect an amount below 30 flow requests per second for the Baseline-Hom trial, and b) 86.49%, 84.83%, and 88.51% for the Baseline-Het trial to connect an amount below 20 flow requests per second. On average, WIMA outperforms the total number of requests/second required by the Baseline rival solution of 86.87% (25.992 out of 197.930,33) in the “Hom” experiments, and 52.66% (14.815,67 out of 31.294,67) for the “Het” experiments.

Furthermore, we also assessed the total amount of time that both WIMA and Baseline solutions spend on setup data flow requests. The flow setup time KPI is an essential prerequisite for an SDN Controller solutions. Figure 7 displays boxplot graphs and cumulative distribution function statistics in an attempt to estimate the agility shown by both the WIMA and Baseline solutions to setup E2E flow requests.

The showcase numerical results of the time spending rate in the Baseline is of 1,736 ms (50 tenants, Figure 7a), 3,477 ms (100 Tenants, Figure 7b), and 4,964 ms (150 Tenants, Figure 7b) for the “Hom” testbed configuration, and 3,047 ms (50 Tenants), 5,961 ms (100 tenants), and 8,847 ms (150 Tenants) for the “Het” experiments. As a result of the seamless MANO approach, on the other hand, WIMA’s trials achieve optimization rates of: 81.28% (325 ms, 50 Tenants), 85.73% (496 ms, 100 Tenants), and 86.34% (678 ms, 150 Tenants) for the “Hom” testbed configuration. With regard to the “Het” testbed configuration, the optimization rate that WIMA achieves over the Baseline is 52.35% (1,452 ms, 50 Tenants), 51.28% (2,904 ms, 100 Tenants), and 53.85% (4,085 ms 150 Tenants). On average, WIMA outperforms the time taken by the Baseline rival solution of 85.27% (499.66 ms out of 3,392.33 ms) in the “Hom” experiments, and 52.72% (2,813.66 ms out of 5,951.66 ms) for the “Het” experiments, which denote added remarkable performance. WIMA achieves much greater agility than the Baseline rival solution by directly invoking the SDN Controllers (in their Northbound APIs) to enforce its intended state in the federated multi-domains. By contrast, the Baseline solution spends more time because its architectural design requires for the OrchFlow abstraction to trigger each in-domain OrchFlow agent, which then orchestrates its respective SDN Controller.

The CDF graphs of Figure 7d reveal that the probability of WIMA-Hom trials requiring response times below 500 ms to setup end-to-end flows is 62.93%, whereas, in the case of the WIMA-Het trials, it is probable that 92.10% will be needed to spend less than 2000 ms during tests with 50 simultaneous tenants. In the same scenario, the Baseline-Hom will likely 1,500 ms in 76.45% of the experimental time, while the response time for Baseline-Het is 94.81%, which is greater than 2000 ms in terms of the E2E Flow Setup Time KPI. During the experimental set with 100 simultaneous active tenants (Figure 7e), WIMA-Het spend 83.83% of the

Table 2: Summary of the results of the assessments

Solution	Simultaneous Tenants	E2E Flow Setup Time (avg)	E2E Flow Setup Time (std)	Number of E2E Flow activations per second	Total E2E activated flows
WIMA-Hom	50	325 ms	345 ms	170.05	153,047
	100	496 ms	514 ms	239.56	215,601
	150	678 ms	684 ms	250.16	225,143
WIMA-Het	50	1,452 ms	379ms	34.62	31,164
	100	2,904 ms	593 ms	34.41	30,974
	150	4,085 ms	770 ms	35.27	31,746
Baseline-Hom	50	1,736 ms	323 ms	28.88	25,994
	100	3,477 ms	452 ms	28.72	25,852
	150	4,964 ms	498 ms	29.03	26,130
Baseline-Het	50	3,047 ms	642ms	16.39	14,749
	100	5,961 ms	980 ms	16.72	15,050
	150	8,847 ms	1,279 ms	16.27	14,648

time remaining below 3,000 ms, whereas the Baseline probably achieves 83.63% and spends more than 5,000 ms. While WIMA-Hom remains 75.36% of the time below 1,000ms, the probability rate for Baseline-Hom is 85.27% for more than 3,000 ms. Finally, in the test settings when there were 150 simultaneous active Tenants (Figure 7f), the Baseline-Het trials had a probability rate of 92.53% for connecting E2E Flows and spending more than 7,000 ms, while WIMA-Het remained at 87.98% of the experimental time staying below 5,000 ms. On the other hand, in the Baseline-Hom trials, there is a probability of 82.27% that its response time will be more than 4,500 ms, while WIMA-Hom spends 83.36% of the time remaining below 1,500ms.

4.1. Analysis of Results

Table 2 summarizes the numerical outcomes collected during the set of experiments for both WIMA and the Baseline trials, as well as the statistics calculated for the E2E flow setup time and the total number of successfully activated E2E data flows when 50, 100, and 150 simultaneous active Tenants are running.

In our experimental evaluations, it was found that during a time of 900 seconds, the WRK benchmarking tool could accomplish, on average, 86.87% (around 197,930.33 in WIMA-Hom and 25,992 in Baseline-Hom) and 52.66% (around 31,294.67 in WIMA-Het and 14,815.67 in Baseline-Het) more flow connections with WIMA than with the Baseline solution. Moreover, WIMA trials also had gains of 84.34% and 52.80% in performance compared with the Baseline experiments by allowing around 83MB and 7MB more data to be transferred within the "Hom" and "Het" testbed configurations, respectively, while spending significantly less time (84.45% in "Hom" testbed, and 52.49% in the "Het" scenario) to accomplish the task. When combined with CDF statistics (based on the total number of requests/second and time response/sec), the WIMA showcases had a higher probability of accomplishing flow requests per second than Baseline, for instance, more than 70 flows per second in 82.99% of the "Hom" testbed configuration with 50 active tenants.

With the feature-matching algorithm's aid, WIMA-Hom achieved best the most remarkable performance of all compared solutions in the experiments. This can be explained by WIMA's ability to process a higher number of simultaneously active tenants without the need for additional abstraction layer instances running in multi-domains, as in the agent-based approach of the rival Baseline solution. By carrying out the more feasible task of mapping the Northbound API of the matching SDN Controller only once, the IC-API component is used to trigger the multi-domains in a straightforward manner. Thus, our WIMA proposed solution provides strong evidence of the suitability and feasibility of deploying a central MANO logic atop underlying Cloud-Network federated multi-domains, as well as featuring homogeneous or heterogeneous SDN Controller solutions (in contrast with agent-based state-of-art solutions).

The remarkable performance of WIMA's seamless MANO features can be attributed to the Global Topology Viewer component capabilities, that maintain a knowledge and view of the global topology of the entire Cloud-Network federation (with UpToDate attributes about nodes, links, resources, and SDN Controllers of the multi-domains). In addition, it computes end-to-end SDN paths in the best possible way and only needs to take a single request into account to trigger the IC-API. Moreover, the WIMA central MANO logic is able to explore the federation domain more efficiently and enforce the multi-domains of SDN Controllers to install state in a parallel paradigm with significantly lower time consumption.

Therefore, it can be concluded that WIMA outperforms differs from the Baseline rival solution through the ability by adopting an optimal approach to enable end-to-end communications across Cloud-Network federated multi-domains when considering either homogeneous or heterogeneous multi-domains structures, which is desirable for the upcoming 5G ecosystems.

5. Concluding Remarks and Recommendations for Future Work

This work introduces WIMA, which provides a modular architecture and is designed with MANO features to connect end-to-end data flows spanning Cloud-Network federated SDN multi-domains seamlessly. The WIMA approach runs as an abstraction layer on top of federated SDN multi-domains, and plays critical MANO role within the entire federation, by making SDN Controllers lightweight enough to enforce indicated state. WIMA provides a global view topology of the entire federation, and allows access to external management applications through a common Northbound API. By separating the functionalities into specialized components that interwork so that it is possible to directly deal with the SDN Controllers, WIMA designs an ontology-based knowledge map for efficient accomplish Northbound API translation without the need for intermediary supporting agents.

We adopted a virtual Cloud-Network federation infrastructure to validate the list of features in the WIMA architecture, and assess the seamless MANO's effectiveness and performance of the seamless MANO with multi-vendor SDN controllers, with regard to the most significant rival state-of-the-art Baseline solution. The outcomes confirm that external management applications harness a common Northbound API to request WIMA seamless MANO with multi-vendor SDN controllers. Moreover, WIMA's effectiveness is confirmed by the fact that it is able to connect around 52.66% (in "Het") and 86.87% (in "Hom") more E2E data flows than the rival Baseline solution. Moreover, a varying number of simultaneous active Tenants (50, 100, and 150) is set for large-scale experiments by stressing data flow connections during an experimental time of 900 seconds. Finally, the same set of large-scale experiments provides proof of WIMA's remarkable improvement in performance compared with the Baseline solution by enabling 52.72% (in "Het") and 85.27% (in "Hom") more agile MANO abstractions to connect E2E data flows across the federated SDN multi-domains, seamlessly.

An complete appraisal of the assessments provides evidence of suggests the suitability and feasibility of WIMA's central MANO common framework since it is able to efficiently enforce the network state in multi-domains of SDN Controllers within a Cloud-Network federation and with a significantly lower time rate. It can thus be concluded that WIMA abilities outperform from the Baseline rival deployment through its ability to provide an optimal solution for the upcoming 5G ecosystems., and it can be distinguished as an alternative solution for the upcoming 5G ecosystems.

The findings obtained from the research endeavors of this paper, including weaknesses and limitations of the proposal developed in the study, indicates the following recommendation areas for further work: (*i*) to enhance improve the WIMA architecture by adding new capabilities to turn it fault tolerant and avoid malfunction features to turn it into a reliable (e.g., by improving availability and survivability); and, (*ii*) to integrate the WIMA architecture into our lab-premises LSDC platform prototype) to strengthen the NECOS' cloud-network slicing concept with the seamless support of a heterogeneously-structured SDN control-plane of federated multi-domains; and finally, (*iii*) to exercise the seamless MANO features of WIMA in a real testbed for obtaining an accurate evaluation of running real-world applications, and pave the way to a next level ecosystem.

Acknowledgment

This research was partially supported by the H2020 4th EU-BR Collaborative Call, under the grant agreement no. 777067 (NECOS – Novel Enablers for Cloud Slicing), funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC.

References

[1] A. Gupta, R. K. Jha, A survey of 5g network: Architecture and emerging technologies, IEEE Access 3 (2015) 1206–1232.

[2] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, O. Rana, The internet of things, fog and cloud continuum: Integration and challenges, Internet of Things 3-4 (2018) 134 – 155.

[3] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, O. Koufopavlou, Software-Defined Networking (SDN): Layers and Architecture Terminology, RFC 7426, 2015. URL: `https://rfc-editor.org/rfc/rfc7426.txt`. doi:`10.17487/RFC7426`.

[4] C. Trois, M. D. DEL FABRO, L. C. E. DE BONA, M. Martinello, A survey on sdn programming languages: Toward a taxonomy, IEEE Communications Surveys Tutorials 18 (2016) 2687–2712.

[5] L. Chouhan, P. Bansal, B. Lauhny, Y. Chaudhary, A survey on cloud federation architecture and challenges, in: R. K. Shukla, J. Agrawal, S. Sharma, N. S. Chaudhari, K. K. Shukla (Eds.), Social Networking and Computational Intelligence, Springer Singapore, Singapore, 2020, pp. 51–65.

[6] M. Carmo, F. S. D. da Silva, A. J. V. Neto, D. Corujo, R. L. Aguiar, Network-cloud slicing definitions for wi-fi sharing systems to enhance 5g ultra dense network capabilities, Wireless Communications and Mobile Computing 2019 (2019).

[7] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. Masoud Sadjadi, M. Parashar, Cloud federation in a layered service model, J. Comput. Syst. Sci. 78 (2012) 1330–1344.

[8] A. Celesti, F. Tusa, M. Villari, A. Puliafito, How to enhance cloud architectures to enable cross-federation, in: 2010 IEEE 3rd International Conference on Cloud Computing, 2010, pp. 337–345.

[9] F. S. D. Silva, M. O. O. Lemos, A. Medeiros, A. J. V. Neto, R. Pasquini, D. Moura, C. E. Rothenberg, L. Mamatas, S. L. Correa, K. V. Cardoso, C. Marcondes, A. Abelém, M. R. Nascimento, A. Galis, L. Contreras, J. Serrat, P. Papadimitriou, NECOS project: Towards lightweight slicing of cloud federated infrastructures, in: 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, Montreal, QC, Canada, June 25-29, 2018, 2018, pp. 406–414. URL: `https://doi.org/10.1109/NETSOFT.2018.8460008`. doi:`10.1109/NETSOFT.2018.8460008`.

[10] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for sdn? implementation challenges for software-defined networks, IEEE Communications Magazine 51 (2013) 36–43.

[11] Ryu, Ryu SDN Framework, 2018. URL: https://osrg.github.io/ryu/, [Online] Accessed 4 Jan 2018. Available at: https://osrg.github.io/ryu/.

[12] ONOS, ONOS - A new carrier-grade SDN network operating system designed for high availability, performance, scale-out, 2018. URL: https://onosproject.org/, [Online] Accessed 4 Jan 2018. Available at: https://onosproject.org/.

[13] OpenDaylight, Project OpenDaylight, 2018. URL: https://www.opendaylight.org/, [Online] Accessed 4 Jan 2018. Available at: http://www.projectfloodlight.org/.

[14] ONF, The Openflow Switch Specification – Version 1.5.0, 2016. [Online] Accessed 4 Mar 2016. Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

[15] J. B. Silva, F. S. D. Silva, E. P. Neto, M. Lemos, A. Neto, Benchmarking of mainstream sdn controllers over open off-the-shelf software-switches, Internet Technology Letters 3 (2020) e152.

[16] A. Hakiri, A. S. Gokhale, P. Berthou, D. C. Schmidt, T. Gayraud, Software-defined networking: Challenges and research opportunities for future internet, Computer Networks 75 (2014) 453–471.

[17] W.-Y. Huang, T.-Y. Chou, J.-W. Hu, T.-L. Liu, Automatical end to end topology discovery and flow viewer on sdn, in: Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, WAINA '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 910–915. URL: http://dx.doi.org/10.1109/WAINA.2014.140. doi:10.1109/WAINA.2014.140.

[18] S. Salsano, P. L. Ventre, F. Lombardo, G. Siracusano, M. Gerola, E. Salvadori, M. Santuari, M. Campanella, L. Prete, Mantoo - a set of management tools for controlling sdn experiments, in: Proceedings of the 2015 Fourth European Workshop on Software Defined Networks, EWSDN '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 123–124. URL: http://dx.doi.org/10.1109/EWSDN.2015.79. doi:10.1109/EWSDN.2015.79.

[19] J. Schultz, R. Szczepanski, K. Haensge, M. Maruschke, N. Bayer, H. Einsiedler, Opengufi: An extensible graphical user flow interface for an sdn-enabled wireless testbed, in: Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, 2015, pp. 770–776. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.113.

[20] A. Shalimov, S. Nizovtsev, D. Morkovnik, R. L. Smeliansky, The runos openflow controller., in: EWSDN, IEEE Computer Society, 2015, pp. 103–104. URL: http://dblp.uni-trier.de/db/conf/ewsdn/ewsdn2015.html#ShalimovNMS15.

[21] A. Voellmy, J. Wang, Y. R. Yang, B. Ford, P. Hudak, Maple: Simplifying sdn programming using algorithmic policies, in: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13, ACM, New York, NY, USA, 2013, pp. 87–98. URL: http://doi.acm.org/10.1145/2486001.2486030. doi:10.1145/2486001.2486030.

[22] A. G. L. C. Stuart Clayman, Feancesco Tusa, Wim on-demand – a modular approach for managing network slices, in: IEEE Conference on Network Softwarization (NetSoft), IEEE, 2010.

[23] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, S. Shenker, Onix: a distributed control platform for large-scale production networks, in: Proceedings of the 9th USENIX conference on Operating systems design and implementation, OSDI'10, USENIX Association, Berkeley, CA, USA, 2010, pp. 1–6.

[24] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, Nox: Towards an operating system for networks, SIGCOMM Comput. Commun. Rev. 38 (2008) 105–110.

[25] T. Choi, B. Lee, S. Kang, S. Song, H. Park, S. Yoon, S. Yang, Iriscoman: Scalable and reliable control and management architecture for sdn-enabled large-scale networks, J. Netw. Syst. Manage. 23 (2015) 252–279.

[26] S. Hassas Yeganeh, Y. Ganjali, Kandoo: A framework for efficient and scalable offloading of control applications, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN '12, ACM, New York, NY, USA, 2012, pp. 19–24. URL: http://doi.acm.org/10.1145/2342441.2342446. doi:10.1145/2342441.2342446.

[27] K. Phemius, M. Bouet, J. Leguay, Disco: Distributed multi-domain sdn controllers., CoRR abs/1308.6138 (2013).

[28] Y. Ganjali, A. Tootoonchian, Hyperflow: A distributed control plane for openflow., in: A. Akella, N. Feamster, S. G. Rao (Eds.), INM/WREN, USENIX Association, 2010. URL: http://dblp.uni-trier.de/db/conf/nsdi/inm-wren2010.html#GanjaliT08.

[29] M. Frate, M. K. Marczuk, F. L. Verdi, OrchFlow: An Architecture for Orchestration of Multiple Controllers in OpenFlow Networks, Journal of Network and Systems Management (2018).

[30] B. Vengainathan, A. Basil, M. Tassinari, V. Manral, S. Banks, Benchmarking Methodology for Software-Defined Networking (SDN) Controller Performance, RFC 8456, 2018. URL: https://rfc-editor.org/rfc/rfc8456.txt. doi:10.17487/RFC8456.

[31] M. Koerner, C. Gaul, O. Kao, Evaluation of a cloud federation approach based on software defined networking, in: 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), 2015, pp. 657–664.

Emídio P. Neto is a researcher at the LaTARC Research Lab and Substitute Professor of Computer Networks at Federal Institute of Education, Science, and Technology of RN (IFRN), Brazil. He is currently pursuing his Master's in Computer Science at the Federal University of Rio Grande do Norte (UFRN). His research interests include Software-Defined Networking, Network Management and Orchestration.

Felipe S. Dantas Silva is a Professor at the Federal Institute of Education, Science, and Technology of Rio Grande do Norte (IFRN), Brazil. He is currently pursuing his Ph.D. in Computer Science at the Federal University of Rio Grande do Norte (UFRN). He is the Research Team Lead of the LaTARC Research Lab and a member of the Research Group in Future Internet Service and Applications (REGINA). He was a visiting researcher at the Network Research Group (NRG) of the Department of Computer Science at the Saint Louis University (SLU), USA, in 2020. His research interests include SDN, NFV, 5G, Mobility Management, Cloud/Fog Computing, Network/Cloud Slicing, QoS/QoE. He is also an IEEE member.

Lucas M. Schneider is a researcher at the LaTARC Research Lab. He is currently pursuing his B.Tech. degree in Computer Networks at the Federal Institute of Education, Science, and Technology of Rio Grande do Norte (IFRN), Brazil. His current main research interests include Software-Defined Networking, Network Management and Orchestration.

Augusto Neto is Associate Professor at the Informatics and Applied Mathematics Department(DIMAp) of the Federal University of Rio Grande do Norte(UFRN), member of the Instituto de Telecomunicações (Aveiro, Portugal), researcher of the National Council for Technological and Scientific Development (CNPq), and leader of the Research Group on Future Internet Services & Applications (REGINA). He got his Ph.D. at the University of Coimbra (2008), and coordinates/participates in research projects funded by national and international development agencies. He is active member of the IEEE Communication Society and participates as editor/reviewer of important journals in the field of computer networks, and has authored and co-authored more than 150 papers. His current research interests are in 5G, slicing, SDN/NFV, Cloud/Edge/Fog Computing, IoT, and Smart Cities.

Roger Immich is an Assistant Professor at the Digital Metropolis Institute (IMD) of the Federal University of Rio Grande do Norte (UFRN). He received his Ph.D. in Informatics Engineering from the University of Coimbra, Portugal (2017). He was a visiting researcher at the University of California at Los Angeles, United States (UCLA) in 2016/2017, and a postdoctoral researcher at the Institute of Computing of the University of Campinas (UNICAMP) in 2018/2019. His research interests are in Smart Cities, IoT, 5G, Quality of Experience, as well as Cloud and Fog computing.