



A WAN Infrastructure Manager Prototype for QoS-based Multi-Site SDN-Enabled NFV Orchestration

Molka Gharbaoui¹ · Barbara Martini² · Piero Castoldi¹

Received: 18 July 2024 / Revised: 31 March 2025 / Accepted: 2 May 2025
© The Author(s) 2025

Abstract

The combination of Network Function Virtualization (NFV) and Software Defined Networking (SDN) enables the deployment of network functions as software components across interconnected telco clouds. Although this approach facilitates the dynamic and efficient management of these functions according to the guidelines of the ETSI NFV management and orchestration (MANO) framework, it brings new challenges in terms of service requirements fulfillment and QoS satisfaction. In this paper, we start by providing an overview of Network Service orchestration and its essential features within multi-site ETSI NFV infrastructures. Subsequently, we introduce a QoS-based WAN Infrastructure Manager (Q-WIM), an open-source prototype of a WIM designed for SDN-based networks. Q-WIM enables the orchestration and deployment of end-to-end Network Services in multi-site NFV environments. In addition to adhering to the standard, Q-WIM advances the State-of-the-Art by incorporating QoS-based capabilities to deploy network services with new features suited for complex scenarios (i.e., bandwidth isolation, delay guarantees). The proposed Q-WIM was implemented and experimentally tested within an emulated virtualized environment with realistic configurations, demonstrating its efficiency in terms of response time and fulfillment of QoS requirements.

Keywords ETSI · Multi-site · WIM · SDN · NFV · QoS

1 Introduction

Upcoming fifth generation and beyond (5 G/6 G) networks are promising to meet the diverse demands of emerging services and applications, ranging from ultra-low latency tasks such as tactile internet and autonomous vehicles, to high-speed,

✉ Molka Gharbaoui
molka.gharbaoui@santannapisa.it

¹ Scuola Superiore Sant'Anna, Pisa, Italy

² Universitas Mercatorum, Rome, Italy

high-capacity communications like augmented reality, and ultra-reliable services such as V2V wireless coordination [1]. In this evolving landscape of modern networking, traditional networking paradigms are being replaced by innovative solutions that harness the capabilities of virtualization (i.e., Network Function Virtualization (NFV)) and software-defined principles (i.e. Software Defined Networking (SDN)) [2]. Furthermore, network providers can leverage flexible network deployments through network slicing, a crucial feature of SDN-enabled NFV that allows for the creation of isolated virtual networks tailored to specific service requirements [3].

As these transformative technologies gain momentum, the need for an adaptive and scalable Wide Area Network (WAN) infrastructure manager becomes increasingly evident, especially in scenarios involving multiple distributed sites. This research paper addresses the gap in the current networking landscape by presenting a novel solution tailored to the challenges of multi-site SDN-enabled NFV orchestration. Our focus is on providing an open-source platform that consolidates the management of network functions. In addition to being fully compliant with the ETSI NFV Management and Orchestration (MANO) architectural framework [4], our prototype supports a set of QoS capabilities that ensure a better performance of network slices in terms of bandwidth and delay guarantees.

In this regard, under NFV MANO, a service can be described as a forwarding graph where two or more network functions are connected by one or more network flows in the network [5]. Those forwarding graphs have to be deployed by appropriately orchestrating the computing, storage, and networking resources [6]. Such operation is ensured by the NFV Orchestrator (NFVO), which manages the virtualized infrastructure and coordinates the deployment of the services across different telco clouds (i.e., Points of Presence (PoPs)) or within one PoP but across multiple resources [7]. Despite the availability of open-source NFVO implementations [8] and that several orchestration solutions have been demonstrated as compliant with ETSI NFV framework [9][10][11], typically the addressed use-cases are pretty simple (e.g., set-up of basic network configurations for connectivity among PoPs [12] and/or consideration of a deployment with a single PoP [13]) and lack several features fundamental for the efficient management of end-to-end services across multi-domains (e.g., abstracted view to NFVO, QoS capabilities). Moreover, no open-source implementation of the WAN Infrastructure Manager (WIM) is currently available, which further restricts the ability of researchers to experiment with multi-site service deployments.

This work focuses on the scenario where more than one PoP is involved in the orchestration process, and where those PoPs are interconnected by an SDN-enabled WAN. In addition to ensuring the connectivity, there is an increasing need to provide QoS capabilities (e.g., bandwidth, bit rate) that allow the enforcement of specific on-demand QoS features (e.g., latency) in a flexible and adaptable way.

Throughout the paper, we discuss the reasons behind developing an open-source QoS-based WAN infrastructure manager (Q-WIM), examining its pivotal features and contributions to network orchestration. We go into details of the dynamic interaction between SDN and NFV, emphasizing how our Q-WIM improves the deployment, management, and optimization of virtualized network functions across

geographically distributed environments. Most importantly, our proposed component is enhanced with a set of attributes that enable the implementation of specific on-demand QoS features in a flexible and adaptable manner for creating network slices within a multi-PoP scenario.

Below we summarize the contributions of this paper:

- We give an overview on Network Service Orchestration in Multi-Site NFV infrastructures and detail the current ETSI specifications in the field;
- We present Q-WIM, an ETSI-compliant WIM that allows to deploy end-to-end Network Services through the provisioning of network connectivity. It supports also the enforcement of specific QoS features such as the bandwidth and delay in order to better meet services requirements especially in complex scenarios;
- We detail the architecture, functionalities, and implementation aspects of the proposed WAN infrastructure manager, highlighting its role in addressing the complexities and challenges of multi-site networking scenarios;
- We present in detail the extensions we performed to the standard to handle new features related to complex scenarios (i.e., bandwidth isolation, delay guarantees);
- We assess the performance of our prototype in terms of appropriate performance metrics such as the overall response time and scalability to offer a comprehensive perspective on the practical implications and benefits of our open-source solution.

The rest of the paper is organized as follows. First, we give a background on ETSI NFV management and orchestration, and we provide a review of the related work in Section. 2. Next, in Section. 3 we focus on network service orchestration in multi-site infrastructures and show its characteristics and current efforts for its standardization. In Section. 4, we present the design of our multi-site SDN-enabled WIM component and specify its features and interfaces, while focusing on the QoS capabilities that it provides. In Section. 5 we detail the steps we followed for the implementation of our prototype. Next, we evaluate the performance of the WIM through a Proof-of-Concept in Section. 6. Finally, we summarize our findings and draw some conclusions in Section. 7.

2 Background and Related Work

2.1 ETSI NFV Management and Orchestration

The combination between NFV and SDN allows to deploy network functions (e.g., router, firewall) as software components running in virtual machines (i.e., VNFs) and to dynamically interconnect them via software network controllers [14]. This approach leads to a significant increase in the flexibility and adaptability of the service platforms. In particular, it makes possible to dynamically compose and orchestrate VNFs and virtual links to deploy end-to-end (E2E) Network Services (NSs) [15]. The resulting virtual assets consist in logically independent infrastructure

partitions, referred to as *network slices*, composed of dynamically-established virtual network, compute and storage capabilities tailored to serve vertical applications with different requirements [2].

In such a scenario, the adoption of some degree of automation is key for convergent slice resource management and service delivery operations. For this purpose, the European Telecommunications Standards Institute (ETSI) developed the NFV Management and Orchestration (MANO) framework, depicted in Fig. 1. The approach mainly consists in decoupling the levels of (i) NFV Infrastructure (NFVI), i.e., the pool of physical/virtual computing, storage and network capabilities; (ii) VNFs, i.e., the pool of individual network functions running in virtual machines deployed in the NFVI; (iii) and end-to-end NSs, i.e., combination of connected VNFs in a self-consistent network service [16]. Along with the functional architecture, the ETSI NFV MANO framework also defines a set of reference points that are used for the exchange of information between the different functional blocks [16].

ETSI is organized in a number of working groups that tackle different aspects of network virtualization. Among those groups, the ETSI NFV Interfaces and Architecture (IFA) Working Group is in charge of several arguments such as the specification of the functional requirements of NFV-MANO, the performance metrics regarding virtualized resources and the requirements and information modelling of NFV descriptors, to name a few. In particular, NFV-IFA 011 [17] defines the Virtual Network Function Descriptor (VNFD), NFV-IFA 014 defines the Network Service Descriptor (NSD), while NFV IFA-005 describes the reference point, *Or-Vi*, which is used to exchange information between the NFVO and Virtualized Infrastructure Manager (VIM) through the support of several interfaces [18].

Finally, NFV-SOL (SOLutions) specifications propose an implementation-ready solution for the concepts defined in the NFV-IFA documents. They mainly regard

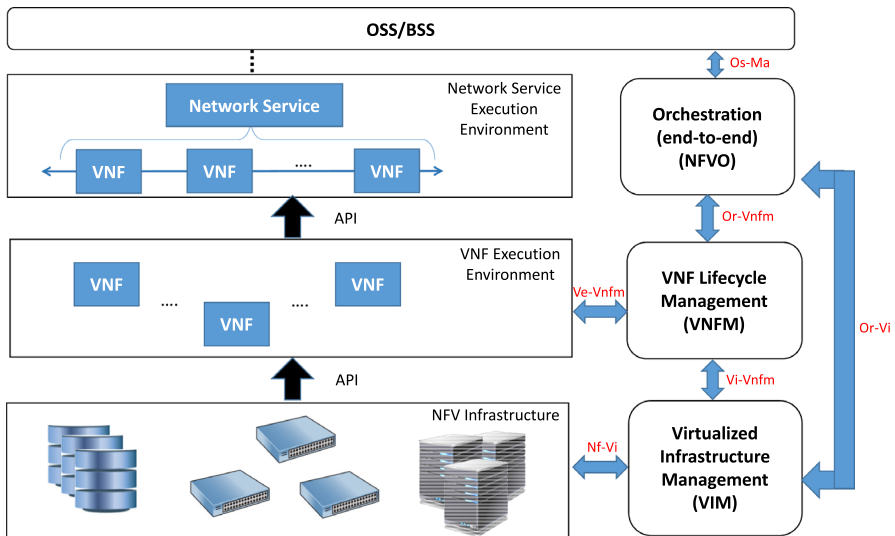


Fig. 1 ETSI NFV management and orchestration framework

the common aspects of the protocols and data models that characterize the NFV-MANO interfaces.

2.2 NFV Orchestrators: Existing tools

Multiple NFV orchestration solutions are available that can be mainly classified into market products or open source initiatives. Open Source MANO (OSM) is a project hosted at ETSI facilities that targets delivering an open source MANO stack closely aligned with the ETSI NFV reference architecture [8]. It plays the role of NFVO and allows for the deployment and management of VNFs through a set of descriptors, which require a certain expertise to be used. The Open Network Automation Platform (ONAP) project is hosted by the Linux foundation and aims to enable end-to-end service agility across multiple domains using a unified platform for NFV and SDN orchestration that is fully compliant with ETSI directives[19]. ONAP includes big data and Artificial Intelligence (AI) modules for optimization of policies and automation of network service deployment and management. However, despite its wide set of features, ONAP requires a high amount of resources to be installed and its deployment is very complex with respect to OSM. Finally, Cloudify is an open source cloud orchestration software platform that was further expanded to include NFV-related use cases [20]. Although Cloudify provides an implementation of most of the ETSI NFV MANO functional blocks, it is not fully compliant as it is not compatible with legacy systems.

The above mentioned orchestration platforms operate based on descriptors where the network services are specified in terms of VNF components, instantiation parameters and VNF forwarding graph (VNF-FG) specification. The management of cross-site connectivity in these orchestration platforms is delegated to the WIM, which however is still partially defined and no open-source implementation of it is currently available.

2.3 Related Work

In the last years, several research works tackled the NFV topic trying to address numerous of its aspects such as provisioning, reliability and scalability [21]. However, the integration of NFV with WAN connectivity still remains a quite unexplored area. [12] presents a WAN virtualization framework, which allows for dynamic end-to-end bandwidth provisioning while leveraging SDN. The authors mainly expose a single virtual switch that can be directly programmed using OpenFlow. This work is however limited to the WAN and the interactions with cloud domains or an NFVO to handle the connectivity are not considered. In [22], a WAN Infrastructure Connectivity Manager (WICM) is proposed to handle the integration of virtualized network functions with WAN connectivity services to compose end-to-end network services. The WICM component focuses on the management of WAN resources while offering a simplified abstracted view of the underlying infrastructure to the NFVO. However, it is based on a proprietary software for the orchestration (i.e., T-NOVA platform) which limits its usage in different scenarios that rely on other frameworks.

[23] presents an architecture design that enables orchestration in a multi-site environment. The paper describes both main components and initial interfaces and APIs considered for multi-site 5 G infrastructures. However, the work lacks some practical insights on the implementation perspective of the proposed architecture.

Some works have proposed to integrate their WIM implementation with the present orchestration platforms. In [25], authors propose a Data Plane Broker (DPB), which supports multi-domain connectivity while it can be integrated with OSM through a WIM plugin. However, even if DPB is capable of setting up a WAN in few seconds and performing resource allocation across it, it does not offer a simple and flexible service data model definition. [26] also presents a solution for OSM, which enables the deployment of a NS over multiple domains interconnected through a network domain. The presented implementation also handles the VNFs interconnection inside the domains which further complicates the scenario. With respect to those works, in this paper, we only focus on the WAN domain up to the DCs/cloud gateways, while the VNFs interconnection is out of scope and left to the VIMs. [27] experimentally demonstrates the possibility to integrate a transport-API based WIM with OSM to enable orchestration over optical networks. As in our proposal, the authors leverage on SDN controllers to handle multi-domain network orchestration however, they focus on IP-over-WDM networks while we consider an SDN-based network.

Some European initiatives have also tackled the orchestration issue where VNFs are deployed over multiple NFVI-PoPs interconnected by WANs. The 5GEx project is based on Multi-domain orchestrator (MdO), which allows the orchestration and deployment of end-to-end network and service elements over multiple domains, technologies and operators [28]. With respect to our proposal, the MdO approach requires a hierarchy of SDN controllers which is not a common practice in existing service providers. MATILDA [11] also brings a framework for the design and implementation of E2E services where aspects related to overall lifecycle of applications in a 5 G ecosystem are tackled. However, the focus is mainly given on the vertical applications and their interaction with the NFV orchestrators and OSS/BSS systems. SliceNet [29] aims to design and demonstrate a vertical-oriented 5 G slicing framework focusing on network management and control of end-to-end slicing operations across multiple operator domains. The main focus of the project is in the context of Multi-Access Edge Computing (MEC) and Radio Access Networks while in this work we focus on the SDN use case. The multi-site scenario in NFV infrastructures has been recently considered in the 5 G-Transformer project [30]. The role of the WIM within the 5 G-Transformer architecture has been defined as the entity that manages the resources within the transport domain and interacts with a mobile transport and computing platform to provide an abstract view of the network and instantiate the virtual networking resources [31][32]. The European project 5Growth [10][33] extends the 5 G-Transformer architecture and aims at developing an end-to-end service platform that might tackle several scenarios such as AI/ML-based [34] and eHealth [35] network services. Finally, the 5 G-VINNI is an E2E 5 G facility comprising multiple interworking 5 G Radio Access Network (RAN) core infrastructures and service orchestrators [36]. It is composed of several facility sites across several European countries. This cross-domain orchestration procedure

requires common standard interfaces and information models across those domains that constitutes a real challenge and still has to be addressed [37].

Table. 1 summarizes the Related Works and shows their similarities and/or differences with respect to our work.

3 Network Service Orchestration in Multi-Site Infrastructures: Beyond ETSI Standardization

3.1 Network Service Orchestration in Multi-Site Infrastructures

The ETSI Industry Specification Group (ISG) produced a report that describes the functional architecture and interfaces necessary for the provisioning and management of multi-site Network Services [24]. The main scope of the report is to provide a set of recommendations and use cases that have to be followed in case the network functions of a given network service are deployed in more than one site (i.e., NFVI-PoP).

The selection of the appropriate VNFs and PoPs is performed at the NFVO level through the use of efficient VNF resource placement algorithms that are out of the scope of this paper [38].

ETSI provides a description of how WAN connectivity might be supported and introduces the concept of WIM, which is presented as a specialized VIM responsible for the connectivity establishment between two or more NFVI-PoPs endpoints across a WAN infrastructure, as shown in Fig. 2. As specified in [24], from the NFVO perspective, the WIM is considered as a component that hides the diversity and complexity of the underlying WAN infrastructure and offers to the NFVO a simplified abstracted view of the network resources. Regarding the interfaces support, the interaction between the NFVO and the WIM has to be performed through the *Or-Vi* reference point, which follows the *NFV-IFA 005* [18] interface specification. In fact, on the one hand the ETSI guidelines allow the NFVO to invoke the same operations as for the VIMs when it interacts with the WIM (e.g., Query Virtualized Network Capacity, Allocate Virtualised Network Resource). On the other hand, the WIM might support a variety of southbound interfaces towards the network controllers in order to fulfill the WAN resource requirements that depend on the specific WAN implementation (e.g., SDN-based, optical).

In the following, we clarify how the WIM can be integrated within the MANO framework and communicate with its existing building blocks. More specifically, as a first step the WIM has to communicate to the NFVO the connectivity reachability among the NFVI-PoPs, which, along with the network connectivity information about the endpoints of each site acquired from the VIMs, allows the NFVO to select the right sites for each VNF according to their affinity. The WAN network information is returned from the WIM to the NFVO and mainly contains the network endpoints reachable over the WAN and the alternate paths between endpoints and the network capabilities (e.g., QoS). Once the NFVO has a local view on the overall multi-site connectivity topology, it is capable to deploy network services and perform resource orchestration tasks. In fact, the NFVO first requests the allocation

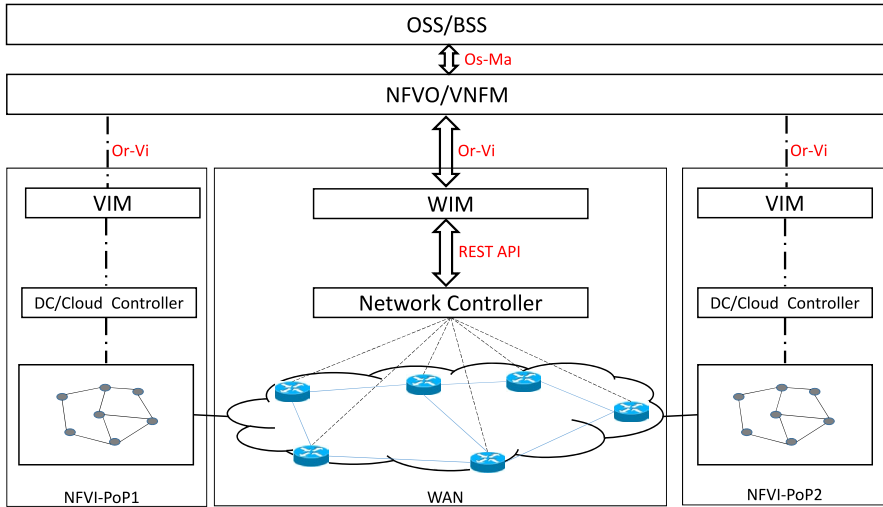


Fig. 2 NFV MANO in a Multi-Site Infrastructure [24]

Table 1 Summary of the Related Work

-	End-to-End resources provisioning	Abstraction to NFVO	OF-based network connectivity	WAN + Cloud domains	Open source platform	Experimental evaluation
Our work	X	X	X	X	X	X
[12]	X	X	-	X	-	-
[13]	X	-	X	-	-	X
[23]	X	X	X	X	-	X
[24]	X	X	-	-	-	-
[25]	X	-	X	-	X	X
[26]	X	-	X	X	X	X
[27]	-	-	-	X	-	X
[28]	X	-	-	X	-	X

of compute/network/storage resources to the VIM(s), then interacts with the WIM to allocate virtualized network resources between the gateways endpoints of the involved NFVI-PoPs with a designated QoS parameters (e.g., bandwidth, latency). The WIM interacts then with the network controller in order to create the connectivity and perform the necessary configurations along the path in the WAN infrastructure (e.g., enforce the flow rules in the switches).

The workflow depicted in Fig. 3 summarizes the interactions between the different entities composing the ETSI NFV framework.

3.2 Current ETSI Specifications for Multi-Site Connectivity

Multi-site connectivity is an active ETSI research topic since it is a major functional requirement for supporting emerging 5 G and beyond use-cases. For this purpose, ETSI released a number of documents that constitute an initial step towards NFV standardization in a multi-site context, namely:

- The NFV-IFA 005 document, which specifies the interfaces supported over the Or-Vi reference point of the ETSI NFV-MANO architectural framework, as well as the information elements exchanged between the NFVO and VIM as shown in Fig. 2 [18]. For each interface, a set of functional requirements is defined.
- The NFV-IFA 032 document, which specifies the interfaces for the management of multi-site connectivity services along with the operations and the information elements that are exchanged over those interfaces [39].
- The NFV-SOL 017 document, which reports about the network connectivity protocols and data model solutions that are available in the industry and analyses the extent to which these solutions address the specified requirements in ETSI GS NFV-IFA 032 and ETSI GS NFV-IFA 005 for the management and support of multi-site connectivity services.

However, despite the architectural design presented by ETSI, further enhancements and optimizations are still needed to build scalable and flexible frameworks. More specifically, the current standard specifications do not define a distinct WIM interface. In fact, due to the heterogeneous nature of the underlying networks and the complexity of end-to-end orchestration, WIM’s functionality and APIs have not been clearly defined yet. Instead, multi-site connectivity

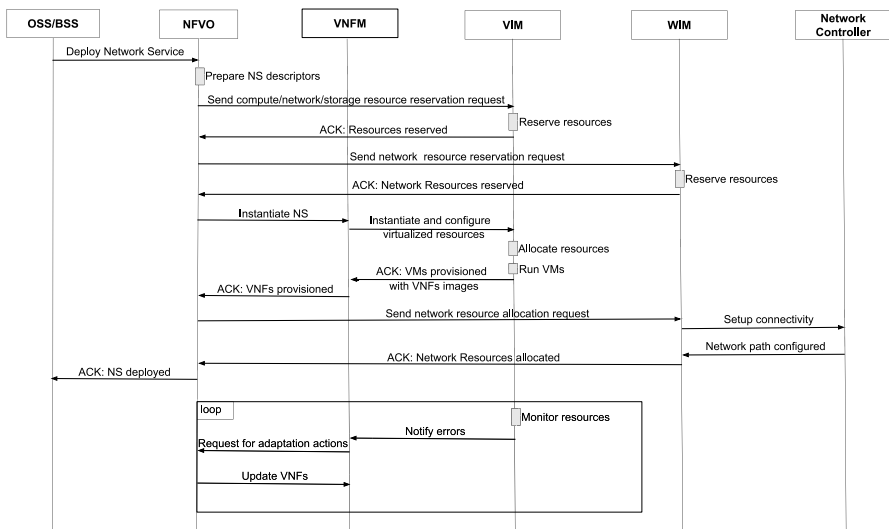


Fig. 3 Workflow for the ETSI NFV framework components [24]

is considered as a special use case where VIM interfaces and data models are inherited. This constitutes an issue since inter-domain interaction architectures have their own specificities and requirements, which differ from intra-domain ones.

Moreover, currently, the framework interfaces are proprietarily implemented based on common aspects of Representational state transfer (REST or RESTful) APIs specified in ETSI NFV-SOL 017, which are not sufficient to cover all the aspects of multi-site interconnectivity. In particular, the data model features for the WIM interface should be further explored and enriched with more elements to cover all the aspects of multi-site scenarios. For example, the bandwidth attribute is currently partially supported while other parameters that are strictly related to the WIM interface are missing such as QoS metrics (e.g., bit rate, delay, packet loss) and networking operations (e.g., addressing, directionality, protection schemes).

Finally, ETSI NFV architecture lacks a clear definition of the abstraction concept of network resources, which instead constitutes a main feature of current network stakeholders and promotes the scalability and modularity of any network service. In fact, currently, the WIM specification exposes an abstraction of the underlying network connectivity to the consumers to facilitate the provisioning and monitoring operations. However, few insights are given on its actual implementation [40]. Moreover, to the best of our knowledge there are currently no mature implementations of WIM modules that could be used to support large-scale provisioning scenarios among multiple NFV sites.

Considering all the aspects described above, the role and implementation features of the WIM remain unclear. This paper contributes to fill the gap with the ETSI standardization by presenting the Q-WIM, which, in addition of being fully compliant with the ETSI NFV architectural model, is further extended with a set of QoS-based capabilities. More specifically, in this work, we follow the ETSI-IFA specifications while considering a multi-site NFV orchestration scenario and we propose the Q-WIM, which aims at reducing the operational complexity and deployment time of end-to-end network services.

4 Q-WIM: WIM for QoS-based Multi-Site SDN-Enabled NFV Orchestration

In this section, we present our *Q-WIM* component, which is a WIM extended with a set of specific features that allow to deploy accurate QoS-aware network services. Furthermore, Q-WIM enables the orchestration of the resources within an SDN-based transport network, and is fully in line with the ETSI MANO framework presented in Fig. 2.

4.1 Q-WIM Functionality

Q-WIM is an open source component [41] in charge of performing the connectivity between two or more NFVI-PoPs through an SDN-based WAN domain. More specifically, Q-WIM implements the following functionalities related with the NFVI connectivity services: i) Northbound API to be used by the NFVO (e.g., provision of network abstracted view, enforcement of logical links); ii) Path computation according to QoS input parameters; iii) Southbound API to SDN controllers to configure the underlying network thus triggering the connectivity over the physical network (e.g., enforcement of forwarding rules).

The Q-WIM is responsible for orchestrating and managing WAN resources by providing an abstracted, high-level view of the network infrastructure. As depicted in Fig. 4, it models the WAN by representing its various components (i.e., transport links, OpenFlow switches) as a set of logical links interconnecting NFVI PoP gateways, allowing for seamless integration with the NFVO. More specifically, the Q-WIM abstracts the complexities of the underlying SDN network elements and presents the WAN as a set of logical transport resources (i.e., virtual network in Fig. 5) rather than a complex mesh of physical connections. It collects topology information from the SDN controller including physical links, nodes, and available bandwidth and then, instead of exposing all the raw details, translates the underlying network elements into virtualized transport resources (e.g., L2/L3 VPNs, Ethernet circuits, or virtual transport tunnels). These resources are then exposed to the NFVO allowing it to request connectivity, bandwidth guarantees, or traffic engineering policies without needing to understand the physical constraints of the network. The WIM can dynamically manage network slices, virtual paths, and end-to-end service chains, optimizing traffic flows based on performance metrics, SLAs, and real-time

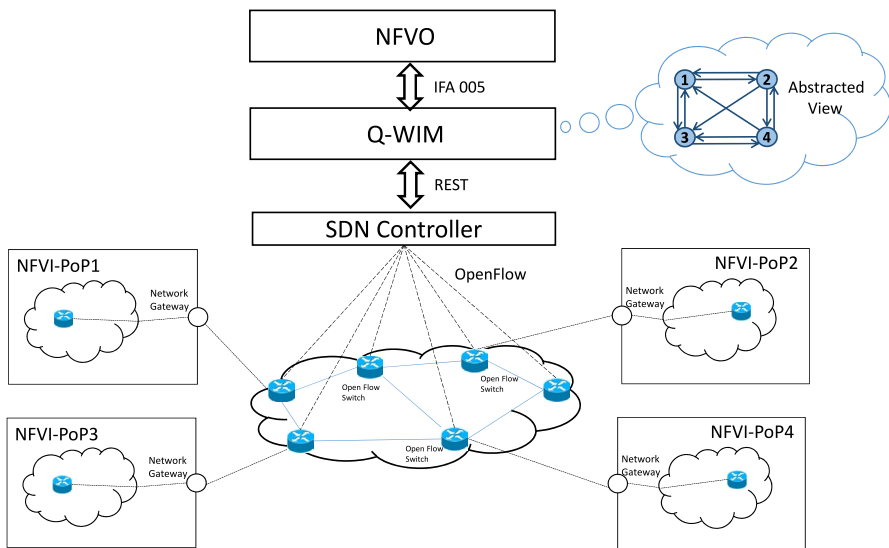


Fig. 4 Q-WIM Abstracted View

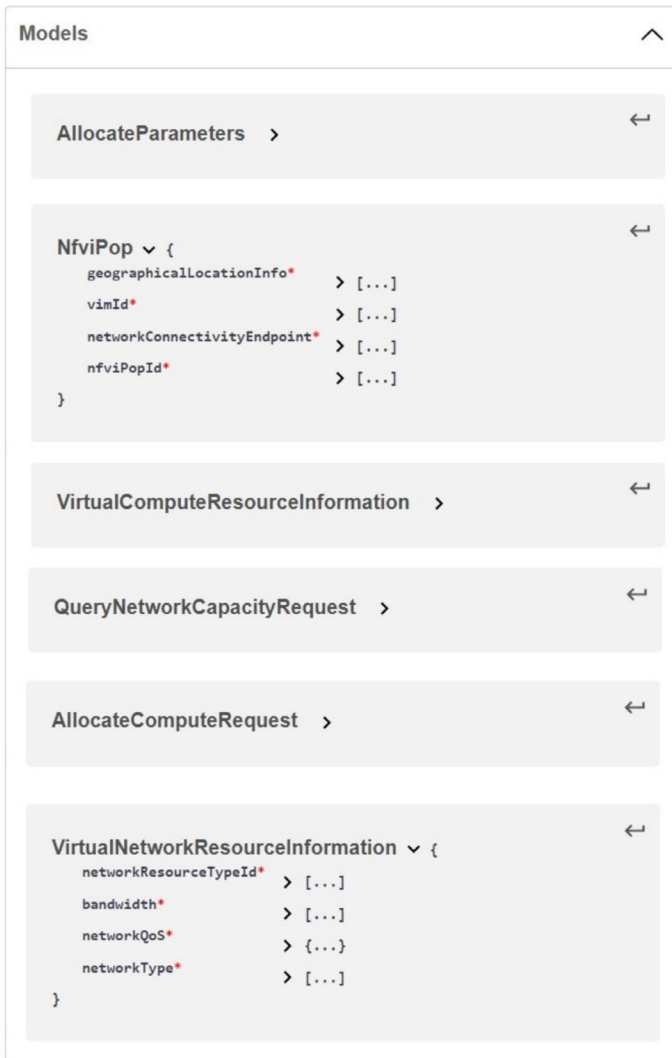


Fig. 5 Q-WIM Abstracted View - Models

network conditions. This abstraction simplifies network operations, enhances scalability, and improves overall network performance.

More specifically, Q-WIM exposes an IFA-005 compliant API to the NFVO to setup the connectivity for network services, query for the actual status of the network resources and allocate part of them or release some of the resources already allocated. In particular, upon request, the Q-WIM can provide an abstracted view to the NFVO that hides the specific characteristics of the underlying infrastructure by representing the WAN as a set of logical links that interconnect NFVI-PoPs gateways. By logical link we refer to a point-to-point connectivity with certain characteristics

between available PoPs. The abstracted view is then composed of these logical links characterized with performance parameters (e.g., latency, bandwidth, cost). The logical link abstraction is technology agnostic (e.g., independent from the fact that the logical link is an optical link or an MPLS link) and the logical link parameters provided to the NFVO are mainly the source and the destination nodes, a measure of the link capacity (e.g., available rate), and latency. Each logical link connects two NFVI-PoPs in a unidirectional way.

In Fig. 4, we show an example of an Q-WIM abstracted view provided, upon request, to the NFVO. In this example, we consider four NFVI-PoPs that are interconnected to each others through a transport network (i.e., SDN network). The Q-WIM only exposes the logical links that interconnect the NFVI-PoPs and does not provide any detail regarding the WAN domain (i.e., technology adopted, switches characteristics (e.g., number, interconnections), traffic steering features). The aim then is to appropriately configure the network to make the traffic flow from one NFVI-PoP to the other under specific QoS conditions and in a transparent manner.

The Q-WIM is triggered once it receives a request for a logical link between two specific NFVI-PoPs which reflects the necessity to appropriately steer the traffic between two VNFs situated in two different locations. Each logical link is uniquely identified by a *logicalLinkId* and characterized by the *srcGwIPAddress* and *dstGwIPAddress* parameters which determine the IPv4 addresses of the NFVI-PoP gateway endpoints (according to the data transmission direction), in addition to the amount of bandwidth to be reserved. Those parameters are then transparently translated into a set of instructions which are directly communicated to the underlying SDN controller. Other parameters not foreseen by the ETSI standard are also provided by the Q-WIM to handle more complex scenarios such as bandwidth isolation among Network Services or delay guarantees. Those parameters will be detailed in Section. 4.2.

Once received, the request is parsed by the Q-WIM and the requirements necessary for the configuration of the logical link are extracted. Then, according to an internal database, the Q-WIM first checks for the availability of the resources. If enough resources are found, it sends a request to the SDN controller to setup a path between the specified NFVI-PoPs while enforcing any specified QoS requirements. Such operation is triggered through the southbound APIs that are exposed by the Q-WIM to the SDN controller which is in charge of managing the underlying network. Once the SDN controller receives the request for the actual setup of the logical link, it directly interacts with the network nodes (e.g., OF switches) to adequately configure them.

Along with the enforcement of the flow entries in the switches, the VIMs managing the NFVI-PoPs involved in this logical link setup also perform the necessary configurations in order to allow the correct steering of the traffic from one VNF to the other. In this work, we focus on the gateway-to-gateway configuration while the intra-NFVI-PoP configurations are performed by the VIM and are considered out of the scope of this paper.

Figure 6 illustrates the different interactions between the NFVO, the Q-WIM, the SDN controller and the data plane as explained above. It is worth pointing out that once the Q-WIM receives the request for resources allocation, it checks for the availability of the resources, and then waits for the acknowledgement from the SDN

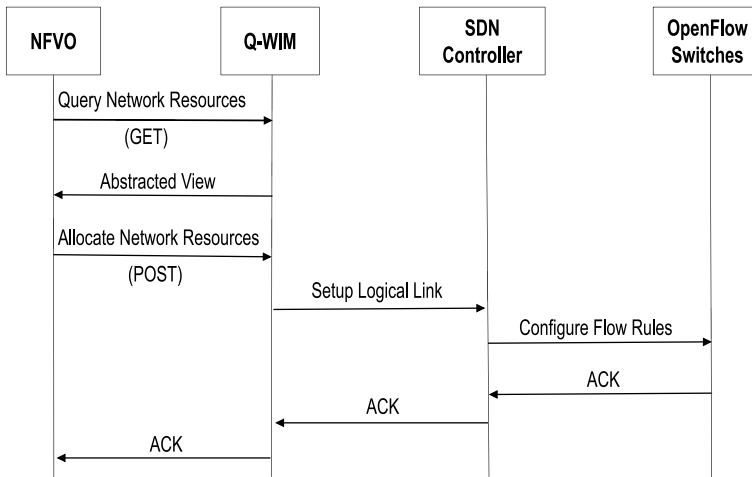


Fig. 6 Q-WIM Workflow

controller before also acknowledging the NFVO. In this way, we guarantee that the NFVO has always an updated and correct view on the status of the logical links and resources. If something goes wrong during the configuration of the network nodes, the NFVO immediately knows that the request setup was not successful.

4.2 Q-WIM QoS Capabilities

With respect to the standard features exposed by any ETSI-compliant WIM, our proposed Q-WIM is enhanced with a set of parameters that allow it to handle novel advanced mechanisms and support new capabilities. Among the new features, Q-WIM handles a "QoS policy" that adds new parameters to the Allocation operation, which represent low level information such as maximum burst size, minimum burst size and maximum bit rate [42]. Those parameters are then enforced by the network controller during the configuration of the data plane. For this purpose, also the underlying SDN controller has to be extended to be able to implement those parameters among the created slices. An example of an SDN slicing application for OF switches can be found in [43]. Another example regards the provision of performance guarantees in terms of network bandwidth and delay [44]. More specifically, delay-related parameters can be considered for the extension of the Allocate Resource Operation to serve the needs for slice-specific delay requirements and bandwidth guarantee. Those parameters might include several attributes such as minimum/maximum bandwidth, marked delay, transmission delay and congestion delay to enable delay guarantees. Also in this use case, some extensions are needed over the control plane, at the level of the network controller SDN applications.

Table 2, summarizes the operations of the IFA-005 compatible interface that are exposed by our Q-WIM component also considering the set of parameters that can be easily added to extend the WIM component.

Table 2 Summary of the interfaces exposed by Q-WIM

Operation	Method	Parameters	Description
Query Virtualised Network Resource()	GET	None	Get the set of logical links with their available bandwidth
Allocate Virtualised Network Resource()	POST	source gateway, destination gateway; max/min bandwidth; max/min burst size; maximum bit rate; m_delay; t_delay; c_delay	Allocate a logical link between two NFVI-PoPs, gateways
Terminate Virtualised Network Resource()	DELETE	Id of the logical link to be deleted	Release the network resources previously allocated

4.3 Q-WIM Interfaces Design Principles

The Q-WIM exposes two interfaces: an IFA-005 compliant northbound interface to communicate with the NFVO, and a RESTful southbound interface to interact with the SDN controller. More details on these two interfaces, extended to serve the above capabilities, are provided in the following.

4.3.1 Extended Northbound Interface

In this work, Q-WIM exposes to the NFVO an IFA-005 compliant interface, which supports three operations that are detailed in the following: *Query Virtualised Network Resource()*, *Allocate Virtualised Network Resource()* and *Terminate Virtualised Network Resource()*:

- *Query Virtualised Network Resource()*: it supports the retrieval of information for the virtualised network resources characterising the WAN. Through this oper-

Table 3 JSON representation of the query virtualised network resource

Request method: GET
URL: /network-resources

Response body:

```
[{
  "networkResourceId": "string",
  "networkType": "string",
  "bandwidth": 0,
  "networkPort": [{
    "networkId": "string",
    "portType": "string",
  }],
  "networkQoS": [
    { "qoSName": "string",
      "qoSValue": "string"
    } ] ] }
```

ation, Q-WIM provides an abstracted view composed of the set of logical links and their current available bandwidth. The Query operation relies on the *REST GET* method with the URI set to `/network-resources`. Table 3 shows an overview of the query. The response body includes an array of the network resources (i.e., logical links) and their parameters such as the id, the available bandwidth on each logical link, the ports ids and their types and the network QoS value (e.g., latency, delay, maximum burst size).

Table 4 JSON representation of the allocate virtualised network resource

Request method: POST

URL: `/network-resources`

Request body:

```
{
  "networkId": "string",
  "networkType": "string",
  "segmentType": "string",
  "ingressPointIPAddress": "string",
  "ingressPointPortAddress": "string",
  "egressPointIPAddress": "string",
  "egressPointPortAddress": "string",
  "locationConstraints": "string",
  "metadata": "string",
  "policy": {
    "policyId": "string",
    "maxCapacity": "string",
    "maxBurstSize": "string",
    "minBW": "string",
    "maxBW": "string",
    "delayflag": "string",
    "mdelay": "string",
    "tdelay": "string",
    "cdelay": "string"
  }
}
```

Response body:

```
{
  "networkId": "string",
  "networkType": "string",
  "segmentType": "string"
}
```

- *Allocate Virtualised Network Resource()*: allows requesting the allocation of virtualised network resources. This operation triggers the allocation of the network resources over one or more virtual links used for deploying a specific network service. The network resource allocation relies on the *REST POST* method with the URI set to `/network-resources`. As shown in Table. 4, the request message carries its content on the body of the *POST* method, which necessarily contains the following parameters: the identifier of the logical link to be allocated (`networkId`), the ingress point IP address (i.e., source gateway), egress point IP address (i.e., destination gateway), ingress point port, egress point port, bandwidth. Other parameters can be added to respond to specific requirements such as delay guarantees. If the logical link is setup correctly, Q-WIM sends a successful response to the NFVO (i.e., code 200) along with the Id of the logical link, and then also updates the database of the resources (e.g., set the new amount of bandwidth that is actually available).
- *Terminate Virtualised Network Resource()*: terminates the allocation of the resources identified by the input parameter of this operation. The de-allocation of the network resources on the selected virtual links uses the *REST DELETE* method with the URI set to `/network-resources/networkId`. The request message also includes the logical link Id in the URL as query string, which identifies the specific virtualized network resource to terminate, while the body of the method is empty. Similarly to the *Allocate* operation, a successful response (i.e., code 200) contains the identifier of the virtualized network resource successfully terminated. At the end of this operation, Q-WIM also updates the database that contains the status of the resources. Table. 5 shows the parameters required for the *Terminate* operation.

4.3.2 Extended Southbound Interface

The southbound interface allows the interaction between the Q-WIM component and the SDN controller that manages the underlying network. More specifically, it allows to dynamically enforce the forwarding rules on the data plane devices (i.e.,

Table 5 JSON representation of the terminate virtualised network resource

Request method: DELETE
URL: <code>/network-resources/{networkId}</code>
Response body:
<pre>{ "networkId": "string" }</pre>

switches and routers) according to the parameters specified in the NFVO request. In this work, we adopt a RESTful interface, which uses HTTP requests to GET, POST and DELETE data [45]. On the other hand, the SDN controller adopts OF as a protocol to configure the network devices. It is worth noting that other protocols such as Netconf could be used for the southbound interface. However, we believe that OF is more in line with the concept of network programmability and orchestration, which constitutes the subject of this paper. Indeed, we consider that Netconf has been previously used for device configuration (e.g., setting of network port addresses) and we adopt OF for dynamic traffic steering and flow entries configurations.

More specifically, our southbound interface offers the following operations:

- Setup logical link operation(): it takes as basic parameters the Id of the logical link, the gateways source and destination IPs and the source and destination ports. According to the use case adopted and to the extensions performed at the level of the network controller, this operation is also extended to handle the features related to the performance isolation or the delay guarantees scenarios (e.g., burst size, delay, maximum bandwidth). The SDN controller checks the switches involved in the request, computes a path from the source to the destination and accordingly configures the flow tables of all the switches composing the path. In this paper, we use ONOS as an SDN controller, which offers the Point-to-point intent feature as an easy way to provide connectivity by configuring the OF switches of the WAN [46]. After the configuration of the flow rules, each logical link is characterized by a unique VLAN Id which allows to isolate the traffic flowing in the WAN and belonging to different logical links. VLAN Ids are shared with the NFVO (i.e., *segmentType* parameter in the POST response body) to allow tagging the traffic originated from the VNFs, while intents Ids are internally kept by the Q-WIM to control their lifecycle (e.g., update, withdraw, delete).
- Delete logical link operation(): it allows to release the network resources previously allocated by deleting the logical link. From the id of the logical link, Q-WIM gets the ids of the configured intents and sends them to the SDN controller in order to delete them. During this operation, the controller updates the flow rules of all the switches belonging to the path that goes from the involved source gateway to the destination gateway.

5 Open Source Q-WIM Prototype

5.1 IFA-005 Northbound Implementation

We use the open-source *Swagger Specification* (recently named as OpenAPI), a browser-based editor, to describe and document our Q-WIM API model [47]. The Swagger specification mainly describes how to access a service or a component through its REST API, including what requests it can handle, what responses may be received, and their format. Swagger is REST based, thus the endpoints are presented

as a set of URI's capable to accept the basic commands such as GET, POST, and DELETE. It allows the description of RESTful applications using objects represented in JSON or YAML file formats and has a user friendly documentation generator. More specifically, Swagger framework is composed of a set of tools, namely a code editor, a GUI and a code generator. The code editor provides specific features allowing to edit the YAML or JSON files used to describe the APIs. The Swagger Codegen provides a stub code with several annotations that help developers to write and implement their API descriptions. To do so, different programming languages are made available such as Java, perl, python, etc. Finally, the Swagger GUI automatically generates the API documentation under an HTML format, thus allowing users to directly interact with the API via a web browser [48].

In Fig. 7, we show the generated documentation of the Q-WIM API using the Swagger editor, while Listing. 1 shows an extract of the API definition in JSON format that describes the structure of the request body sent to the operations in the REST API and the structure of any response bodies. The API consists of three request types, where we specify the employed method and the required parameters.

The Q-WIM code is open and currently released in Github [41]. It consists of around 46k lines of Java code.

5.2 Test Environment and Methodology

To validate the performance of Q-WIM, a Proof of Concept environment was setup to provide a realistic scenario where a set of PoPs were interconnected through an SDN-based network composed of OpenFlow switches. More specifically, the performance evaluation of the Q-WIM component has been carried out on a virtual machine with the following characteristics: 1 core of *i7* – 5500U at

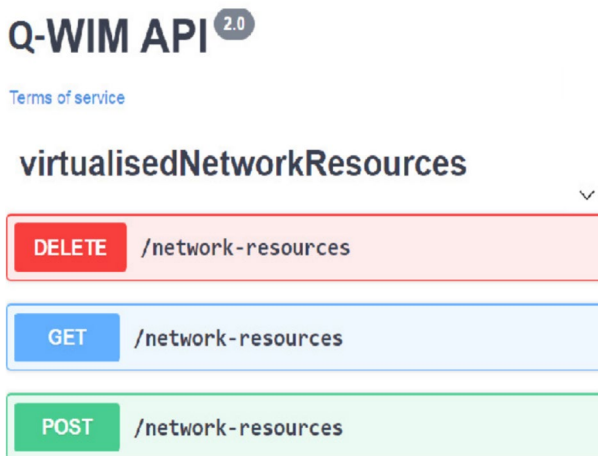


Fig. 7 API documentation for Q-WIM displayed with Swagger UI

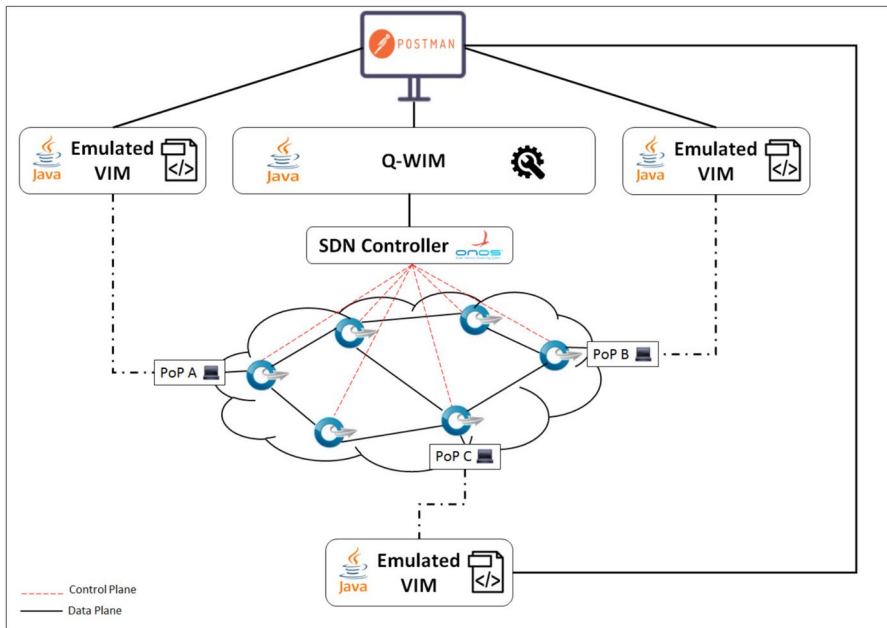


Fig. 8 Experimental Setup

2.40GHz, 4GB of RAM and 30GB of disk storage. As shown in Fig. 8, we considered ONOS (version Uguisu, 2.4.0) [49] as the SDN controller, while the data plane has been emulated using the Mininet tool [50]. Through Mininet, we created a topology composed of a set of OF switches interconnected to emulate the WAN topology. It is worth pointing out that the SDN topology can be implemented either using a cluster of OF virtual switches (OvS) or Mininet. On the other hand, the requests coming from the NFVO have been sent using the Postman tool [51]. Regarding the NFVI-PoPs, we used 3 scripts that emulate the behavior of the VIMs and we considered connectivity requests between one source and one destination nodes attached to two switches in the SDN topology. Each test was repeated 10 times and an average was used as the result for all the plots.

```

paths:
/network-resources:
delete:
parameters:
- name: ids
description: Identifier of the virtualised
network resource(s) to be terminated.
- application/json
responses:
'200':
description: Identifier(s) of the virtualised
network resource(s) successfully terminated.
'400':
description: Bad request
get:
parameters:
- name: networkQueryFilter
description: Query filter based on e.g. name,
identifier, meta-data information or status
information, expressing the type of
information to be retrieved. It can also be
used to specify one or more resources to be
queried by providing their identifiers.
- application/json
responses:
'200':
description: Element containing information
about the virtual network resource(s)
matching the filter.
post:
parameters:
in: body
required: true
operationId: allocateNetwork
- application/json
responses:
'201':
type: string
description: Identifier of the created network
resource.

```

Listing 1: Excerpt of the Swagger definition of the Q-WIM API

Figure 9 shows a Wireshark capture [52] between the different entities composing our framework. In particular, it details the RESTful API messages that are effectively exchanged between the NFVO and the WIM/VIMs. It can be observed that, after the retrieval of the current status of the resources, a request is first sent to the VIMs to configure the intra-PoPs network, then to the Q-WIM to inter-connect the PoPs through the configuration of the SDN network resources. The same interactions are performed for the deletion request.

	No.	Time	Source	Destination	Info
Retrieve compute and network resources availability	409	1308.839245455	NFVO	Enhanced WIM/ VIMs	GET /mtpbase/abstract-resources HTTP/1.1
Configure compute resources at VIM1 and VIM2	410	167.232156374	Enhanced WIM/ VIMs	NFVO	HTTP/1.1 200 OK (application/json)
	416	189.662907590	NFVO	VIM1	POST /mtpbase/abstract-compute-resources HTTP/1.1 (applicati...
Configure intra-PoP network resources at VIM1 and VIM2	418	194.839894996	VIM1	NFVO	HTTP/1.1 200 OK (application/json)
	425	220.713026074	NFVO	VIM2	POST /mtpbase/abstract-compute-resources HTTP/1.1 (applicati...
	427	224.754150176	VM2	NFVO	HTTP/1.1 200 OK (application/json)
	431	235.868683951	NFVO	VIM1	POST /mtpbase/network_resources HTTP/1.1 (application/json)
Configure network resources in the SDN network	433	239.280255757	VIM1	NFVO	HTTP/1.1 200 OK (application/json)
	437	249.568847079	NFVO	NFVO	POST /mtpbase/network_resources HTTP/1.1 (application/json)
	439	251.580942221	VM2	NFVO	HTTP/1.1 200 OK (application/json)
Delete network resources in the SDN network	450	260.961567748	NFVO	Enhanced WIM	POST /network-resources HTTP/1.1 (application/json)
	457	268.733676755	Enhanced WIM	NFVO	HTTP/1.1 200 OK (application/json)
Delete compute resources at VIMs	461	280.489965498	NFVO	Enhanced WIM	DELETE /mtpbase/network_resources?networkId%20=1 HTTP/1.1
	463	280.695858336	Enhanced WIM	NFVO	HTTP/1.1 200 OK
Delete network resources at VIMs	467	290.510443176	NFVO	VMs	DELETE /mtpbase/abstract-compute-resources?computeId=1 HTTP/1.1
	469	292.137969364	VMs	NFVO	HTTP/1.1 200 OK (application/json)
Delete network resources at VIMs	476	303.760712670	NFVO	VMs	DELETE /mtpbase/abstract-network-resources HTTP/1.1 (applica...
	478	304.468182517	VMs	NFVO	HTTP/1.1 200 OK

Fig. 9 Captured RESTful API messages exchanged between the NFVO and the WIM/VIMs

6 Proof-of-Concept

6.1 API Response Time

In this subsection, we evaluate the Q-WIM northbound interface response time, which refers to the time elapsed between receiving a Query, Allocate or Delete message, elaborating it and sending back a response or an acknowledgement. To do so, in each iteration, we repeatedly sent random requests from Postman to the Q-WIM component and recorded the average response times of all requests. The average response time is evaluated while varying the number of OF switches in the WAN topology.

Figure 10 shows the results for the three operations of the northbound interface, namely the Query, Allocate and Delete. More specifically, we can notice that by increasing the number of switches in the WAN, the response time of the Query operation remains stable below 200 ms. In fact, through this operation, the Q-WIM sends the abstracted view of the topology to the NFVO. The view is retrieved from the database and sent under the format of an xml file. It contains the set of logical links and their characteristics and does not depend on the number of network devices. On the contrary, the response time of the Allocate operation strictly depends on the

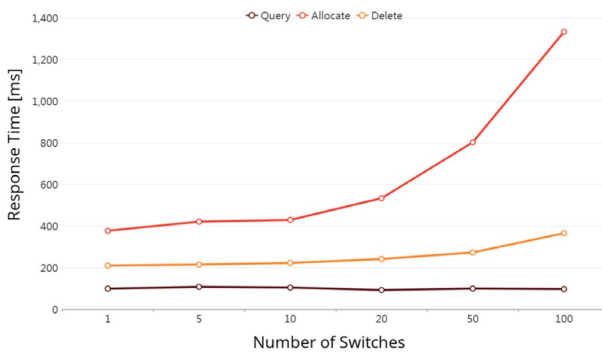


Fig. 10 Northbound Response Time

size of the topology since it allows for the actual setup of the logical links and then the configuration of the OF switches in the network. In the experiment, we have considered that the configuration is performed to interconnect two NFVI-PoPs while involving a random number of switches to be traversed, depending on the path selected by the SDN controller. From the plotted results, we can see that the more switches are present in the topology, the more time is necessary for the setup of a single logical link, which mainly corresponds to the time necessary for the enforcement of the flow entries in the switches. More details are given in Section 6.2. Finally, the response time of the Delete operation also slightly increases by varying the number of switches but in a more contained way with respect to the Allocate case. Also for the Delete, the largest part of the response time is dedicated to the tear-down time of the logical links. However, while during the Allocate, the setup requires to wait for the enforcement of the policies in all the traversed switches, in the Delete, tagging the logical link as "withdrawn" is enough. More details on how this operation is performed are given in Section 6.3.

It is worth highlighting that the overall response time for the three operations is composed of the following events: socket initialization, DNS lookup,

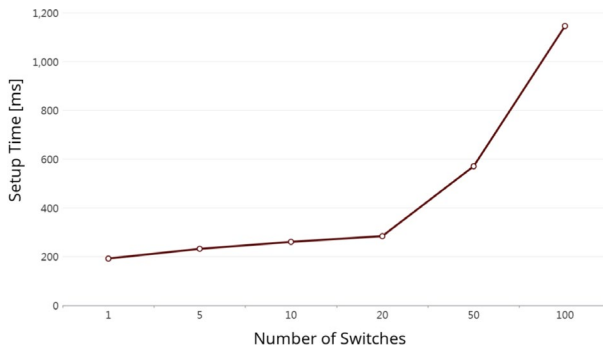


Fig. 11 Network Setup Time

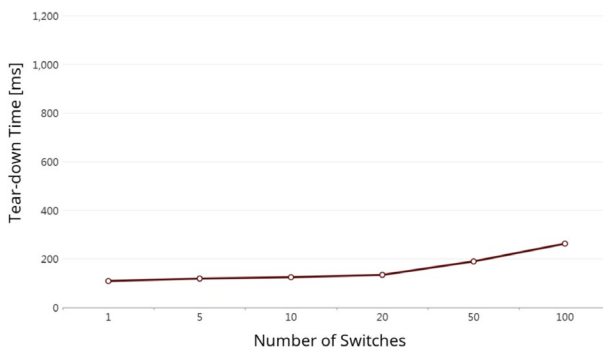


Fig. 12 Network Tear-Down Time

TCP handshake, Transfer start and download. Most of these events take less than 1ms or are even saved in the cache, while almost all the response time is due to the "Transfer" operation in the Allocate and Delete case which refers to the time required by the Q-WIM to actually process the request. In the Query case, the response time is almost equally divided between the "Transfer" and "Download" operations, since the content of a file is transferred in the response.

6.2 Network Setup Time

Figure 11 plots the setup time necessary for the configuration of a unidirectional logical link, while varying the number of switches in the network. This time corresponds to the actual setup of all the necessary flow rules in the switches that belong to the selected path between the gateway source and gateway destination. To do so, we have adopted the *intent* framework [46] provided by the ONOS controller which allows for specifying the network control requirements in form of policies using a high-level abstraction rather than technical configuration details [53]. The intent specifications (i.e., IP and port of the source gateway, IP and port of the destination gateway, QoS parameters) are provided by the Q-WIM to the SDN controller, and then the ONOS core is in charge of translating those requirements into actionable operations such as installing the flow rules on the switches, setting VLAN IDs, etc. In such a way, the network configuration is simpler and vendor and device-agnostic.

From the results, we can notice that the average setup time of one logical link remains below 1 s even when the number of switches in the WAN is around 100. However, we also observe that the setup time increases with the increase of the number of the switches in the topology. This is mainly due to the performance of the ONOS controller that we used in the Proof-of-Concept. More specifically, as the topology expands (more switches, links, and hosts), having only one centralized controller that has to communicate with a larger number of devices

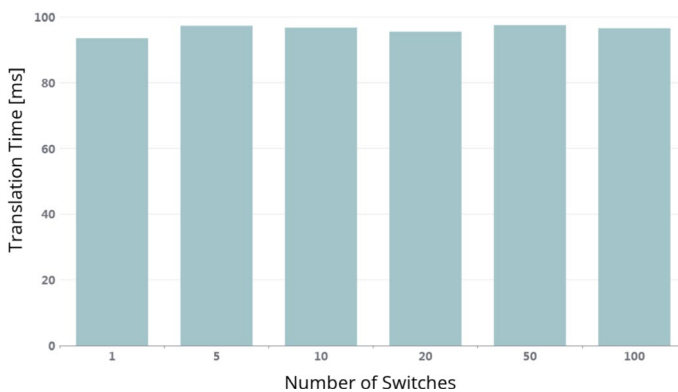


Fig. 13 Q-WIM Translation Time

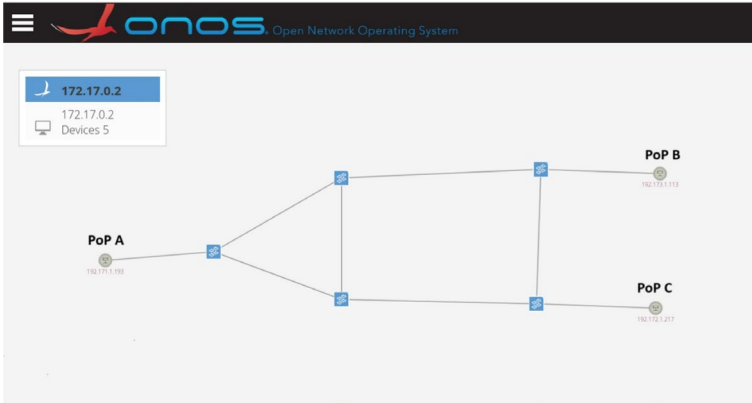


Fig. 14 SDN network topology in ONOS GUI

increases the time required for flow rules configuration. Moreover, ONOS uses shortest path algorithms (i.e., Dijkstra) for route computation. As the number of switches and links grows, the number of possible paths also grows exponentially thus increasing the time required to compute optimal paths before installing the flow rules.

6.3 Network Tear-Down Time

Figure 12 plots the tear-down time which corresponds to the time necessary for deleting the flow rules corresponding to a given logical link in the topology. More specifically, during this operation, the SDN controller removes the intent from the system and deletes the flow entries previously configured. From the results, we can see that the average tear-down time increases as the number of switches in the network increases. As stated in the previous paragraph, this is mainly because of the length of the selected paths in the network increases. However, the tear-down

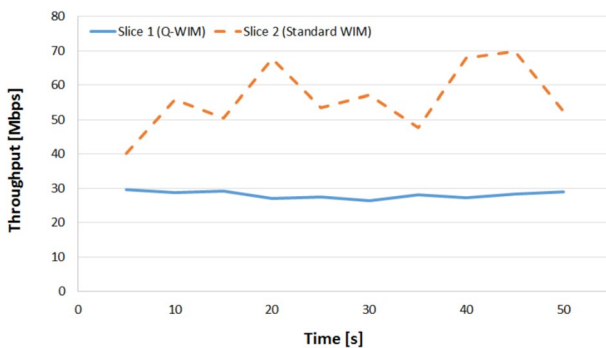


Fig. 15 Performance evaluation: Q-WIM vs. Standard WIM

time is lower than the setup time mainly because while for the setup the controller has to wait for the installation of all the flow rules in order to tag the intent as "installed" and send the intent id to the Q-WIM, in the delete case the controller immediately tags the intent as "Withdrawn" and replies to the Q-WIM then removes the flow rules from the switches.

6.4 Translation Time

In this paragraph, we measure the weight of the translation time with respect to the overall response time needed to setup a logical link. By translation time, we refer to the amount of time spent by the Q-WIM in converting the requirements of the request received from the NFVO into the configuration parameters sent to the SDN controller.

Figure 13 shows the time needed by the Q-WIM to process the incoming connectivity requests before sending the intents installation requests to ONOS. We first note that the average translation time for all the requests is below 100 ms which clearly shows that the time needed by the Q-WIM to extract the logical link requirements from the NFVO request, prepare the request for the SDN controller and then send it is very low with respect to the intent setup time. Moreover, from the results we can clearly state that Q-WIM scales well. In fact, as the number of OF switches increases in the network, the translation time remains almost constant showing that the processing time at the Q-WIM is minimal.

6.5 Throughput

We reproduce the scenario depicted in Figure 8 composed of three PoPs, interconnected using an SDN network infrastructure including OpenFlow Virtual Switches (OVSS) deployed using the Mininet emulator. For this experiment, we consider a smaller topology reported in Fig. 14 to have a common link as a bottleneck (from PoP A to PoP B and PoP C). We set the links capacity in the SDN network to 100 Mbps. The experiment is composed of two phases. In phase one we start with the creation of the first slice. We create a slice between PoP A and PoP B with specific QoS policies. In this experiment, we fix the rate limit to 30 Mbps. After the slice creation, we execute the Ping command between PoP A and PoP B to check if the slice was created successfully. We then generate UDP traffic at saturation rate (around 100 Mbps) between the two PoPs using the iPerf tool. In the second phase, we create a second slice between PoP A and PoP C and validate its coexistence using the Ping command. In this phase, the QoS capabilities of Q-WIM are disabled to emulate the behavior of a standard WIM. We then generate the same amount of UDP traffic as for slice 1 using the iPerf tool.

Results are plotted in Fig. 15 We can notice that while in slice 1 the throughput is almost constant and below the fixed threshold (i.e., 30 Mbps), in slice 2 the traffic almost doubles the bandwidth used since no limitations are enforced. Moreover, the traffic shape in slice 2 is completely different with very high burstiness and instability. In fact, generating traffic at saturation rate with unlimited policies creates network

congestion in the common links among the slices thus leading to delaying and/or dropping packets and then creating fluctuations in the throughput.

The obtained results show the role of Q-WIM in satisfying the QoS requirements that each service demands for each slice in a shared environment. Results also confirm the performance isolation among the deployed slices since the traffic rate of slice 1 is not impacted by the traffic rate of slice 2, which is guaranteed by the Q-WIM showing its role in having policed and/or unpoliced slices running on the shared infrastructure.

7 Conclusions and Future directions

The support of multi-site network service deployment is currently one of the major requirements for 5 G emerging applications and scenarios. In this work, we presented Q-WIM, an open-source solution designed for the orchestration and deployment of end-to-end network services in multi-site SDN-enabled NFV infrastructures.

The reference architecture for our WAN infrastructure manager is the ETSI NFV MANO framework, which was described in detail in this paper to characterize its building blocks and key features for the design and deployment of our component. With respect to the standard, Q-WIM provides a set of QoS features (e.g., bandwidth, delay) that allow the deployment of complex network services within advanced scenarios.

Through rigorous analysis, a deployment in a realistic virtualized environment and performance evaluations, we demonstrated the effectiveness of our proposed WAN infrastructure manager in enhancing network performance, scalability, and response time. As a future work, we plan to enhance the multi-site scenario with further details related to additional heterogeneous services and applications and provide an exhaustive evaluation of the Q-WIM operations under dynamic conditions.

Acknowledgements This work was supported by the EU under the Italian National Recovery and Resilience Plan (PNRR) of NextGenerationEU Partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”). The authors also gratefully acknowledge financial support from Universitas Mercatorum within the framework of the 2024 FIN/RIC program (ID 23-FIN/RIC).

Author Contributions M.G. and B.M. contributed to the conceptualization, design and interpretation of the data of the paper. M.G. contributed to software design and implementation. M.G., B.M. and P.C. contributed to methodology and compilation of the whole paper including drafting the article and revising it critically for important intellectual content. All authors contributed to the approval of the final version.

Funding Open access funding provided by Scuola Superiore Sant’Anna within the CRUI-CARE Agreement.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long

as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Jiang, W., et al.: The road towards 6g: A comprehensive survey. *IEEE Open J. Commun. Soc.* **2**, 334–366 (2021)
2. Ordonez-Lucena, J., et al.: Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Commun. Mag.* **55**(5), 80–87 (2017)
3. Yousaf, F., et al.: NFV and SDN-Key technology enablers for 5G networks. *IEEE J. Select Area Commun.* **35**(11), 2468–2478 (2017)
4. ETSI: Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Architectural Framework Specification, ETSI GS NFV 006 V4.4.1 (2022)
5. Gharbaoui, M., et al.: An experimental study on latency-aware and self-adaptive service chaining orchestration in distributed nfv and sdn infrastructures. *Comput. Netw.* **208**, 108880 (2022)
6. Schardong, F., et al.: NFV resource allocation: a systematic review and taxonomy of VNF forwarding graph embedding. *Comput. Netw.* **185**, 107726 (2021)
7. Vaishnavi, I., et al.: Realizing services and slices across multiple operator domains, in: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, (2018), pp. 1–7
8. OSM, <https://osm.etsi.org/> (2024)
9. Baranda, J., et al.: 5G-transformer meets network service federation: design, implementation and evaluation, in: 6th IEEE Conference on Network Softwarization (NetSoft), (2020) pp. 175–179
10. Li, X., et al.: 5Growth: An end-to-end service platform for automated deployment and management of vertical services over 5G networks. *IEEE Commun. Mag.* **59**(3), 84–90 (2021)
11. Gouvas, P., et al.: Design, Development and Orchestration of 5G-Ready Applications over Sliced Programmable Infrastructure, in: 29th International Teletraffic Congress (ITC 29), Vol. 2, pp. 13–18 (2017)
12. Lara, A., et al.: WAN virtualization and dynamic end-to-end bandwidth provisioning using SDN, in: Optical Fiber Communication Conference (OFC), (2015)
13. Mangués-Bafalluy, J., et al.: 5G-TRANSFORMER Service Orchestrator: design, implementation, and evaluation, in: European Conference on Networks and Communications (EuCNC), pp. 31–36 (2019)
14. Rotsos, C., et al.: Network service orchestration standardization: A technology survey. *Comp. Stand. Interfaces* **54**, 203–215 (2017)
15. Gharbaoui, M., et al.: Experimenting latency-aware and reliable service chaining in next generation internet testbed facility, in: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–4 (2018)
16. ETSI, Network Functions Virtualisation (NFV); Management and Orchestration; Report on Management and Orchestration Framework, ETSI, GR NFV-MAN 001 V1.2.1 (2021)
17. ETSI, Network functions virtualisation (NFV) release 4; management and orchestration; VNF descriptor and packaging specification, GS NFV-IFA 011 4.2.1 (May 2021)
18. ETSI, Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification, ETSI, GS NFV-IFA 005 4.2.1 (May 2021)
19. ONAP, <https://www.onap.org/> (2024)
20. Cloudify, <https://docs.cloudify.co/> (2024)
21. Bouras, C., et al.: SDN & NFV in 5G: Advancements and challenges, in: 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), pp. 107–111 (2017)

22. Carapinha, J., et al.: Deployment of virtual network functions over multiple WAN interconnected pops, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 252–257 (2017)
23. Contreras, L., et al.: Interworking of softwarized infrastructures for enabling 5G multi-site slice orchestration, in: IEEE Conference on Network Softwarization (NetSoft), pp. 458–463, (2019). <https://doi.org/10.1109/NETSOFT.2019.8806656>
24. ETSI, Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services, ETSI, GR NFV-IFA 022 3.1.1 (2018)
25. Simpson, S., et al.: Dataplane broker: Open WAN control for multi-site service orchestration, in: IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–6 (2019)
26. Karamichailidis, P., et al.: Enabling Multi-Domain Orchestration using Open Source MANO, OpenStack and OpenDaylight, in: IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 1–6 (2019)
27. Bravalheri, A., et al.: VNF Chaining across Multi-PoPs in OSM using Transport API, in: Optical Fiber Communication Conference (OFC), (2019)
28. Sgambelluri, A., et al.: A multi-operator network service orchestration prototype: The 5G exchange, in: Optical Fiber Communications Conference and Exhibition (OFC), pp. 1–2 (2017)
29. Wang, Q., et al.: SliceNet: End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks, in: IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), (2018) pp. 1–5
30. Li, X., et al.: Automated service provisioning and hierarchical sla management in 5g systems. *IEEE Trans. Netw. Serv. Manage.* **18**(4), 4669–4684 (2021)
31. Casetti, C., et al.: Network slices for vertical industries, in: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 254–259 (2018)
32. Baranda Hortiguera, J., et al.: Realizing the network service federation vision: Enabling automated multidomain orchestration of network services. *IEEE Veh. Technol. Mag.* **15**(2), 48–57 (2020). <https://doi.org/10.1109/MVT.2020.2979558>
33. Chang, C.-Y., et al.: Performance isolation for network slices in industry 4.0: The 5growth approach, *IEEE Access* **9**, 166990–167003 (2021)
34. Baranda, J., et al.: On the Integration of AI/ML-based scaling operations in the 5Growth platform, in: IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 105–109 (2020)
35. Baranda, J., et al.: NFV service federation: enabling multi-provider ehealth emergency services, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1322–1323 (2020)
36. 5G-VINNI project, [Online]. Available: <https://www.5g-vinni.eu/>
37. Ordóñez-Lucena, J., et al.: Cross-domain slice orchestration for advanced vertical trials in a multi-vendor 5g facility, in: European Conference on Networks and Communications (EuCNC), pp. 40–45. (2020) <https://doi.org/10.1109/EuCNC48522.2020.9200940>
38. Laghrissi, A., Taleb, T.: A Survey on the Placement of Virtual Resources and Virtual Network Functions. *IEEE Communications Surveys & Tutorials* **21**(2), 1409–1434 (2019). <https://doi.org/10.1109/COMST.2018.2884835>
39. ETSI, Network functions virtualisation (nfv) release 3; management and orchestration; interface and information model specification for multi-site connectivity services, ETSI, GS NFV-IFA 032 3.6.1 (January 2022)
40. I. R. 8453, Framework for abstraction and control of TE networks (ACTN), [Online]. Available: <https://www.rfc-editor.org/rfc/pdf/rfc8453.txt.pdf>
41. WIM code repository, <https://github.com/5growth/5gr-ri/tree/master/ri/plugins/WIM/ONOS-OpenFlow-Slicing/onos-plugin> (2024)
42. Cunha, V., et al.: 5Growth: Secure and Reliable Network Slicing for Verticals, in: European Conference on Networks and Communications (EuCNC), (2021)
43. Scano, D., et al.: Network Slicing in SDN Networks, in: Int. Conf. on Transparent Optical Networks (ICTON), (2020)
44. Chang, C., et al.: Slice Isolation for 5G Transport Networks, in: IEEE NetSoft Conference and Workshops (NetSoft), (2021)
45. RESTFul API, <https://restfulapi.net/> (2024)

46. ONOS Intent Framework, <https://wiki.onosproject.org/display/ONOS/Intent+Framework> (2024)
47. Swagger Editor, <https://editor.swagger.io/> (2024)
48. Musyaffa, F.A., et al.: Minimally invasive semantification of light weight service descriptions, in: IEEE International Conference on Web Services (ICWS), pp. 672–677 (2016)
49. ONOS Controller, <https://wiki.onosproject.org/display/ONOS/Downloads> (2024)
50. Mininet Official Site, <http://mininet.org/> (2024)
51. Postman, <https://www.postman.com/> (2024)
52. Wireshark, <https://www.wireshark.org/> (2024)
53. Martini, B., et al.: Intent-based zero-touch service chaining layer for software-defined edge cloud networks. *Comput. Netw.* **212**, 109034 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Molka Gharbaoui is an assistant professor at the Scuola Superiore Sant'anna, Pisa, Italy. She received her Ph.D. degree in Innovative Technologies of Information & Communications Engineering and Robotics in 2012 from the Scuola Superiore Sant'anna, Pisa. Her main research interests are in the field of intent-based networking, software defined networking, network function virtualization and network orchestration. She has been involved in several EU projects and has co-authored more than 70 papers appeared in international journals and conferences.

Barbara Martini is a Full Professor at Universitas Mercatorum, Italy in Computer Engineering and Security. She is Head of "Engineering and Sciences" Department and Coordinator of PhD Program in Big Data and Artificial Intelligence. She is also Affiliate Professor with Scuola Superiore Sant'Anna, Italy. Previously, from 2003 to 2022 she worked at the Italian University Consortium for Telecommunication (CNIT). At CNIT she coordinated research activities since 2009 in the field of network virtualization and orchestration in 5G, network control/management architectures and programmability, service platforms for networking, and security solutions for multi-domain networks and NFV deployments. Current research interests are mainly on 5G/6G network architectures also relying on ML/AI, intent-based networking, distributed learning in decentralized network environments with focus on manufacturing domain. She co-authored 120+ papers in top-notch scientific journals and international conference proceedings.

Piero Castoldi (IEEE Senior Member) is Full Professor in Telecommunications and Director of the TeCIP Institute at Scuola Superiore Sant'Anna (Pisa, Italy), and Responsible for the Networks and Services area. He has been involved with various responsibilities in several national and EU FP7, H2020 and HE projects and he has managed several corporate-sponsored projects with the Italian Railway Infrastructure Company, Ericsson, Telecom Italia and other companies. He has served as project manager for several projects of the Italian National Consortium for Telecommunications (CNIT). His teaching activity has been extensive including the coordination of two Erasmus Mundus Master programs. His most recent research interests lie in the areas of optical network architectures, network management and control, interconnection networks for Data Centers, 5G/6G networking. He is author of more than 500 technical papers published in international journals and international conference proceedings and he has filed more than 20 patents.