vL2-WIM: Flexible virtual layer 2 connectivity services in distributed 5G MANO domains

Timo Kellermann
i2CAT Foundation
timo.kellermann@i2cat.net

Ferran Cañellas
i2CAT Foundation
ferran.canellas@i2cat.net

Ricardo González
i2CAT Foundation
ricardo.gonzalez@i2cat.net

Daniel Camps-Mur
i2CAT Foundation
daniel.camps@i2cat.net

***Abstract*—Future 5G networks will be implemented as distributed clouds where virtual network functions and services are instantiated on demand. To support the required flexibility and automation, novel data center interconnect technology must support stringent data plane requirements brought along by 5G Radio Access Networks (RANs), while delivering the necessary flexibility in the service definition and enabling automation in the service provisioning. In this paper we present vL2-WIM, a novel WAN Infrastructure Manager (WIM) that enables virtual layer 2 services across data centers in distributed 5G MANO deployments. We provide a detailed evaluation of vL2-WIM showing how complex connectivity services composed of up to 30 Virtual Network Functions (VNFs) in different compute domains can be provisioned in less than 15 seconds.**

***Index Terms*—WAN Infrastructure Manager, SDN, NFV, Multi domain**

I. INTRODUCTION

Data Center Interconnect (DCI) is at the core of future 5G networks, which will be implemented as a distributed cloud where most of the mobile network functions are virtualized and can be flexibly instantiated at different locations [1].

ETSI NFV MANO [2] defines the reference framework to manage such a distributed cloud, allowing to provision *Network Services* (NSs) that are composed by a set of concatenated *Virtual Network Functions* (VNFs) connected through *Virtual Links* (VLs).

The previous virtual constructs are enabled by the native virtualization capabilities of the physical infrastructure, composed of x86 servers and *Software-Defined Networking* SDN enabled network equipment. The *Virtual Infrastructure Manager* (VIM) is in charge of virtualizing data center resources, including storage, network and compute. However, when a NS is defined that spans multiple NFV infrastructures (NFVI) (data centers), which is a typical requirement for 5G NSs [7], then the network resources connecting the NFVI facilities also need to be virtualized, which is a capability provided by the *WAN Infrastructure Manager* (WIM) [3].

Mediating between the network services and the virtual resources made available by the VIM and the WIM, we encounter the *NFV Orchestrator* (NFVO), of which OSM [4] and ONAP [5] are the most representative open source alternatives. Fig. 1 depicts the components of a typical distributed 5G MANO system, highlighting the roles of the NFVO, the VIM and WIM in a multi data center deployment, where the base stations (gNBs) connect to NFVI *Points of Presence* (PoPs).

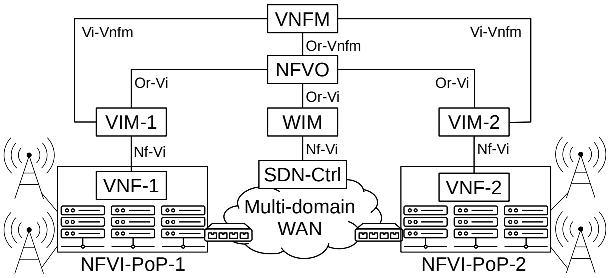


Fig. 1. 5G MANO architecture with NFVO, VIM and WIM.

The main contribution of this paper is the design and evaluation of a novel software defined WIM MANO component, which we refer to as *vL2-WIM*. vL2-WIM allows to provision on-demand virtual layer 2 connectivity services between data centers, while minimizing signaling transmitted across the WAN as well as service start up delay, requiring only SDN enabled carrier Ethernet devices in the data plane, and providing a high level interface that allows to easily configure the topology of the required connectivity services.

Major network vendors promote today VXLAN EVPN as a solution to stretch virtual layer 2 networks across data centers [8]. This solution requires an underlay IP/MPLS network that is not adequate to support certain types of 5G RAN functional splits [9], e.g. fronthaul, and does not provide flexibility in the service definition. Instead, vL2-WIM delivers virtual layer 2 services directly over an SDN enabled carrier Ethernet network, and allows to easily develop novel connectivity services as applications running on an SDN controller. The work in [10] motivates the use of the ONF Transport API (T-API) to allow a MANO WIM to orchestrate multiple optical transport connectivity domains. This work though does not provide a detailed designed of the WIM and does not evaluate the scalability of their solution. In [11] the authors present DataPlaneBroker (DPB), which is, like vL2-WIM, an ETSI MANO WIM offering virtual layer 2 services. The authors integrate DPB with OSMv5 and demonstrate experimentally that DPB can establish WAN connectivity in around 10 seconds, which is the same order of magnitude that we report with vL2-WIM in Section III. Unlike vL2-WIM, DPB does not support a flexible service definition and does not minimize ARP signalling across the WAN. Net2Plan [12] is a multi-tenant infrastructure management tool that can control several

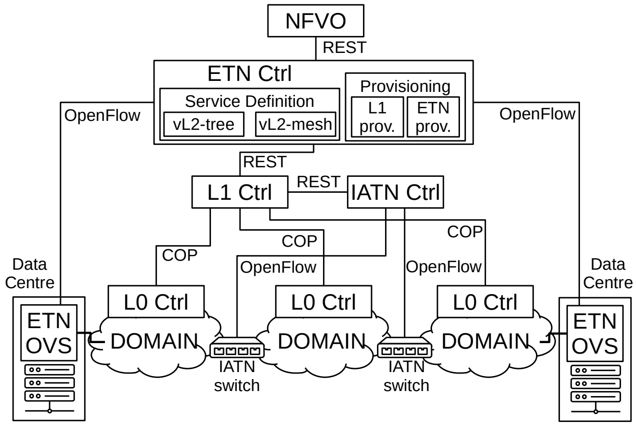


Fig. 2. vL2-WIM architecture with ETN, L1, IATN controller and L0 controllers for each domain. Data center PoPs are connected to domains via ETN software switches. Communication from NFVO to ETN controller and from ETN controller to L1 controller via REST protocol. IATN switches are OpenFlow capable hardware switches while ETN switches are implemented as software function.

VIM instances and provision connectivity across data centers through an ONOS controller. Unlike vL2-WIM, Net2Plan considers the transport network to be composed of only one domain. Finally, the 5G-XHaul project [13] defines a hierarchical SDN control plane to provide connectivity service across multi-domain transport networks. vL2-WIM builds on the hierarchical control plane defined in 5G-XHaul and adds flexible service definition capabilities, and an enhanced ETN pipeline to minimize ARP signalling across the WAN, which is an important feature for 5G *Ultra Reliable Low Latency Communication* URLLC services.

This paper is structured as follows. Section II presents the architecture and the detailed design of vL2-WIM. Section III includes an exhaustive performance evaluation using a custom made testbed that validates the scalability of vL2-WIM. Finally, Section IV summarizes and concludes the paper.

II. vL2-WIM DESIGN

Fig. 2 depicts the vL2-WIM architecture, where we can distinguish three control layers organized in a hierarchical manner. First, the WAN domain is decomposed into multiple domains, each controlled by a *Level 0* (L0) controller. Each L0 domain is composed of *Transport Nodes* (TNs), which may implement different forwarding technologies. For example one domain could comprise standard Ethernet devices, whereas another domain could consist of Ethernet Time Sensitive Network (TSN) devices supporting fronthaul flows. Each L0 controller is responsible for setting up label switched paths (LSPs) within its domain, which in a carrier Ethernet network can be implemented using VLAN tagging. To interconnect different L0 domains and provide an end-to-end path, vL2-WIM considers the *Inter-Area Transport Nodes* (IATN), which are SDN controlled physical switches that can stitch LSPs across L0 domains. The IATN Controller is the control plane entity in charge of programming IATN nodes. The second control layer is composed by the *Level 1* (L1) controller, which interfaces with the L0 and the IATN controller and has an overall view of the WAN. The third and highest control layer is composed by the *Edge Transport Node* (ETN) Controller that controls the ETN nodes deployed at the edge of the network. ETNs are SDN devices located inside the data-center that stitch the traffic of the VNFs of a given NS within a data-center to the end-to-end LSP across the WAN that connects remote data-centers. ETNs support both a purely software or a hardware accelerated implementation as explained in Section II-A.

vL2-WIM computes end-to-end paths across the multi-domain WAN in a hierarchical way. First, the L1 Controller aggregates the end-to-end topology at the network domain level and calculates the shortest path across domains, thus selecting domains and IATNs along the path. Notice that to perform this operation the L1 Controller does not need to know the detailed topology within each L0 domain. Subsequently, L0 controllers compute paths within their respective domains.

To enable communication across the different SDN controllers vL2-WIM leverages the Control Orchestration Protocol (COP) [14], which is based on REST, and allows the L0 controllers to expose towards the L1 controller primitives to request a path and to collect topology. Using COP, the same service primitives, but referring to the WAN, are also made available from the L1 Controller to the ETN Controller. Finally, the ETN Controllers exposes a custom REST interface that allows the NFVO to request allocation of end-to-end paths across data-centers. More details on this interface are provided in Section II-B.

A. Data plane elements and Forwarding Model

vL2-WIM considers a data plane composed of Transport Nodes (TNs) as generic Ethernet forwarding devices in the L0 domains, Inter-Area Transport Nodes (IATNs) between the domains, and Edge Transport Nodes (ETNs) connecting endpoints in the data-center to the end-to-end LSPs across the WAN. TNs and IATNs are implemented using carrier grade Ethernet devices with OpenFlow support, which is widely available across the main networking vendors.

ETNs implement an advanced pipeline, including local ARP resolution, which is an important feature to reduce signalling and initial communication delay in 5G URLLC services composed of VNFs deployed across distributed compute locations. Since the pipeline operations required to enable local ARP resolution are not available in current hardware SDN switches, ETNs feature a hybrid software-hardware implementation, where user plane traffic is accelerated through a standard Openflow switch whereas ARPs are resolved using a software pipeline based on *Open vSwitch* (OVS) [16]. This architecture is shown in Fig. 3.

Fig. 3 depicts the forwarding pipelines for the TN, IATN and ETN devices. The forwarding pipeline in TNs and IATNs is based on OpenFlow and is implemented using a single table. In TNs, packets are forwarded according to their ingress port and outermost VLAN tag, which signal the LSP within the L0 domain. In addition to that, IATNs perform VLAN tag translation to be able to forward packets across domains.

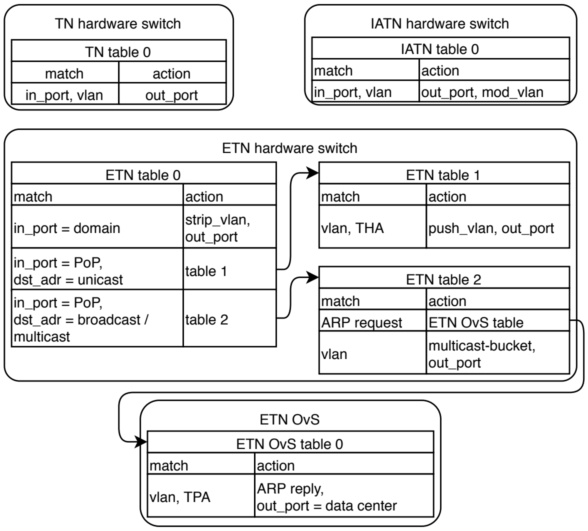


Fig. 3. Forwarding pipelines for TNs, IATNs and ETNs.

Unlike TNs and IATNs, ETNs include a multi-table pipeline in the hardware switch, and a single table to enable local ARP resolution in the software switch. ETN table 0 in the hardware switch separates packets according to ingress port and unicast or broadcast packet type. Packets coming from the data center are then processed in 2 further tables.

Unicast packets are matched in table 1 based on inner VLAN tag and Target Hardware Address (THA), so that the correct transport VLAN tag can be applied to reach the destination ETN. Additionally, the inner VLAN tag, which is used to signal the network service inside the data-center, can be swapped to match the destination data-center VLAN tag if it differs from the source. Notice that the inner VLAN tags will be provided by the VIMs operating in each data-center and are therefore available to the NFVO. Non unicast packets from a VNF are inspected in table 2, and if they are ARP requests, they are forwarded to the ETN software switch, where the ARP resolution table is used to generate ARP responses. Information available in the ARP Request is used to generate the ARP Response. inside the software switch. After matching the remote VNF IP address and modifies the received ARP request. The Source Hardware Address (SHA) of the ARP request is moved to the THA, Source Protocol Address (SPA) is swapped with the Target Protocol Address (TPA). Then, the SHA is filled with the MAC address associated with the remote VNF IP address of the ARP request.

The ETN Controller is the entity in charge of provisioning the bindings between MAC and IP addresses inside the ETN pipeline, when a new service is provisioned. The required information can be extracted from the VIM, deploying the VNFs, and from the base stations participating in the deployed service. Broadcast messages, as well as ARP requests, which cannot be matched in the ETN table, are forwarded via group buckets to all ETNs that have VNFs registered that belong to the same group based on the source VLAN tag (see ETN table 2). Finally, packets coming from the WAN are striped of the transport VLAN tag and forwarded to the data-center (see ETN table 0.

It is worth highlighting that the described ETN pipeline also admits a purely software based implementation using OVS, which can be useful in scenarios where OpenFlow capable hardware switches are not available. In this case kernel bypass techniques such as DPDK [15] can be used to accelerate packet processing.

B. *ETN controller and vL2 service programming model*

Upon processing an NSD including a Virtual Link Descriptor (VLD) across data-centers, the NFVO (c.f. Fig. 2) issues a virtual L2 (vL2) service request to the ETN controller. The VLD included in the NSD specifies the type of vL2 service required to interconnect the VNFs across the data-centers. This vL2 service type is included in the request issued by the NFVO to the ETN controller, where it is linked to a particular vL2 service implementation.

The ETN controller is implemented as an application running on top of the ONOS SDN controller [17] containing the following modules. The *vL2 service* module implements the service logic, which defines the topology of the vL2 service. We have implemented three canonical vL2 services, namely a *vl2-p2p* service connecting two remote data-centers, a *vL2-tree* service, where all leaf VNFs have L2 reachability to a root VNF, and a *vL2-broadcast* service, where all VNFs in the NSD are in the same L2 broadcast domain. The vL2 service module makes use of the *L1 provisioning* and the *ETN provisioning* modules to program the forwarding pipelines of the data-plane elements according to the service logic. Provisioning of L1 Controller and ETN data plane elements into ETN Controller is done via REST API on the respective controllers.

C. *IATN controller design*

The IATN controller is also implemented using ONOS. It contains a database maintaining the mapping between L0 domains and physical interfaces for the IATN devices it controls. The IATN Controller offers a REST interface to the L1 Controller, used to issue service requests that contain source and destination domains with their corresponding LSP identifiers, i.e. VLAN tags. Upon receiving a service request, the IATN Controller looks up the involved L0 domains in its database, identifies the targeted IATN device, and installs the required flow rules using OpenFlow.

D. *End-to-end path establishment*

Fig. 4 illustrates the steps required to set up an end-to-end path across the WAN. First, the NFVO triggers the ETN Controller through its north-bound REST interface. Then, the ETN Controller triggers a COP service call request to the L1 Controller, which generates a COP service call towards all the L0 controllers that need to be traversed to establish the end-to-end path. Requests to L0 controllers are sent in parallel.

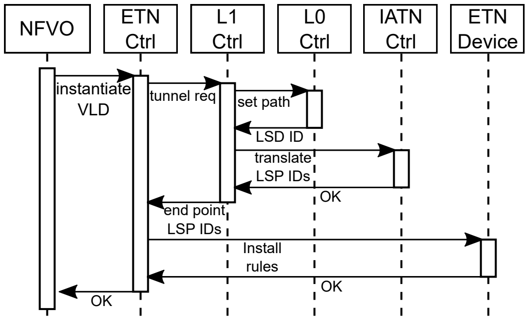


Fig. 4. Traffic sequence between controllers for end-to-end path establishment.

After programming the LSPs in their respective domains, L0 controllers report the used LSP identifier, i.e. VLAN tags, back to the L1 Controller. Armed with the LSP identifiers, the L1 Controller triggers the Inter-Area Transport Node (IATN) Controller to stitch LSPs across domains. Subsequently, the L1 controller reports back to the Edge Transport Node (ETN) Controller the LSP identifiers that need to be applied at each ETN device to bind the internal NS data-center traffic to the end-to-end path. Finally, the ETNs Controller programs the forwarding tables of the involved ETN devices (c.f. section II-A) using the OpenFlow protocol.

In the next section we evaluate the time required by vL2-WIM to provision end-to-end connectivity services when using different connectivity services.

III. PERFORMANCE EVALUATION

To evaluate vL2-WIM, we developed a testbed using two Intel NUC devices with a Intel Core i7-6770HQ 4 core processor that can emulate different WAN topologies. We implement the controller hierarchy depicted in Fig. 2 using separate Virtual Machines (VMs) for the L0, L1, IATN and ETN controllers. The data-plane of the data center and L0 domains is implemented using Linux namespaces and Open vSwitch instances[1]. We use separate interfaces for each VM interconnection, which allows us to monitor the control traffic between the different controllers.

The goal of our evaluation is to benchmark the scalability of vL2-WIM in terms of the overall time and the signalling overhead required to provision a vL2 service. For this purpose, we execute two types of experiments. In the first experiment, we define three vL2 services, namely a *vl2-p2p* service, a *vl2-tree* service, and a *vl2-broadcast* service, and study how provisioning time and signalling overhead scale as the number of VNFs and data centers included in the network service grows from 1 to 30. In the second experiment, we evaluate how vL2-WIM scales when the number of L0 domains between data centers increases from 1 to 8 by again studying the service provisioning time and the signalling overhead.

To measure service provisioning time, we launch an ICMP request in the first VNF that is instantiated in the service and measure the time until the ICMP response is received from the last VNF instantiated in the service. Notice that in this way, we ensure that all data-planes have been configured and traffic can flow between data centers. To measure signalling overhead, we capture inter-controller traffic using *Wireshark*. To ensure that results are statistically significant in experiment 1, we run 30 tests per service size, and in experiment 2, we run 50 tests for each number of domains.

Fig. 5 depicts the three vL2 services used in experiment 1. In the *vl2-p2p* service, a local and a remote data centers are connected to the WAN through a single ETN each, and the number of VNFs instantiated in the remote data center increases. In the *vl2-tree* service, each VNF in the network service is instantiated in a separate data center connected to the WAN through an ETN, and all leaf VNFs have connectivity only to the root VNF. In the *vl2-broadcast* service, the physical topology is the same as in the *vl2-tree* service, but all VNFs can connect to each other thus requiring a full mesh of end-to-end LSPs between ETNs.

A. Service Provisioning Time

Fig. 6(a) depicts the service provisioning time for the *vl2-p2p* (left), *vl2-tree* (middle) and *vl2-broadcast* (right) services as the number of VNFs in the network service grows from 1 to 30. Box plots are used to represent the measured service provisioning times, with the box limits representing the 25% and 75% values of the cumulative distribution function (cdf), and the whiskers the 5% and 95% values.

We can see how the *vl2-p2p* and *vl2-tree* services exhibit a linear growth in service provisioning time, with the *vl2-p2p* service staying below 0.5 seconds and the *vl2-tree* service below 2 seconds in the worst case. The *vl2-broadcast* service exhibits a quadratic growth, which results in up to 15 seconds when the NSD contains 30 VNFs. The main factor driving the service provisioning time is the number of end-to-end LSPs across the WAN that need to be deployed by the L1 controller for each service. For the *vl2-p2p* service only one end-to-end LSP needs to be deployed between the two data centers, which is reused for all the connections between the root VNF and the remote VNFs. All connections between remote VNFs are resolved locally by the ETN in the remote data center. Thus, adding new VNFs to the service requires only provisioning new rules in the ETNs. In the *vl2-tree* service, each new VNF is deployed in a different data center, hence a new end-to-end LSP between the first and the new ETN needs to be deployed, which results in the linear growth in service provisioning times. In the *vl2-broadcast* service, a full mesh connectivity among ETNs is required to provide a single broadcast domain for VNFs deployed across different data centers, which results in $n\times(n-1)$ end-to-end LSPs being deployed for a network service with n VNFs. Notice though that even in this very complex case the service provisioning time of vL2-WIM is only 15 seconds in the worst case, which is very fast for practical deployments. Another advantage of vL2-WIM in the *vl2-broadcast* service is that ARPs between VNFs within the same network service are resolved locally in

---

[1] Using a software ETN in our evaluation does not impact the obtained results that focus on control plane performance and not on data plane.

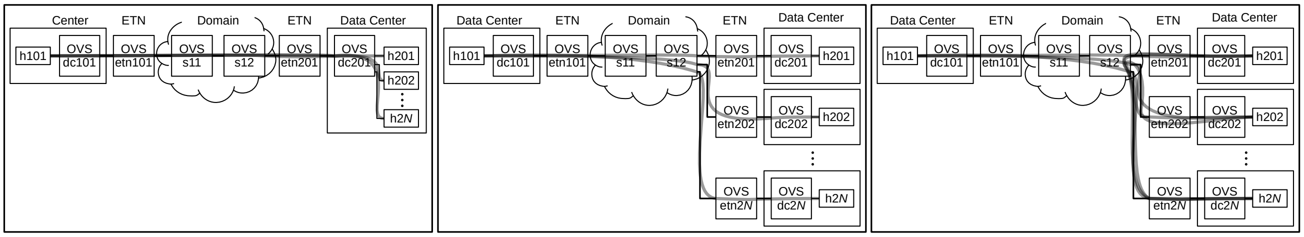
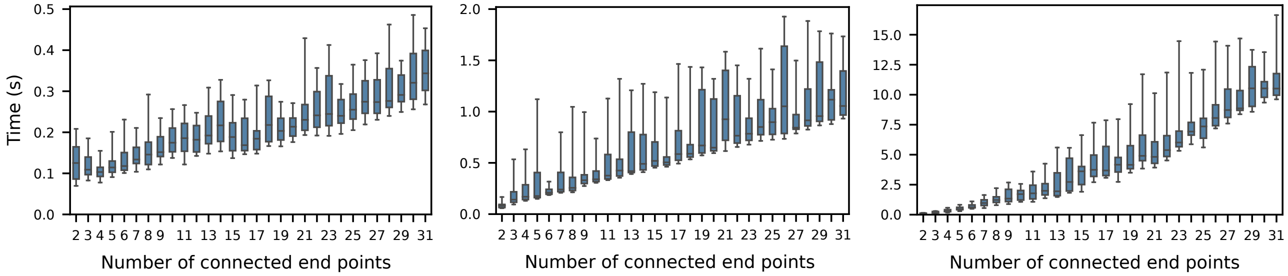
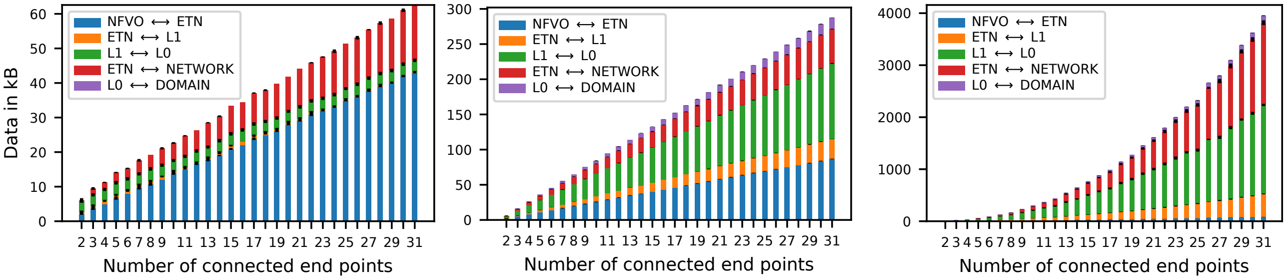


Fig. 5. Services used in experiment 1: *vl2-p2p* (left), *vl2-tree* (middle) and *vl2-broadcast* (right)



(a) Service provisioning times for the single tunnel, vl2-tree and vl2-broadcast services.



(b) Signalling overhead for the single tunnel, vl2-tree and vl2-broadcast services.

Fig. 6. Results for Experiment 1

each ETN, reducing address resolution latency and minimizing WAN traffic.

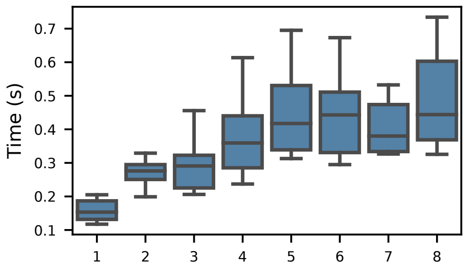
Fig. 7(a) depicts the service provisioning time for the *vl2-p2p* service with two VNFs in the network service with an increasing number of L0 domains in the WAN. We can see that the service provisioning increases up to four domains, but then stagnates for higher number of domains, being in all cases below 0.7 seconds. A clear linear trend is not observed in this case because the L1 controller sends requests for new paths to the L0 domains in parallel.

Analyzing the service provisioning time for different types of vL2 services and large NSD instances (up to 30 VNFs), we conclude that vL2-WIM is sufficiently scalable to add a marginal increase in the provisioning time of practical ETSI NFV network services in real 5G mobile networks.

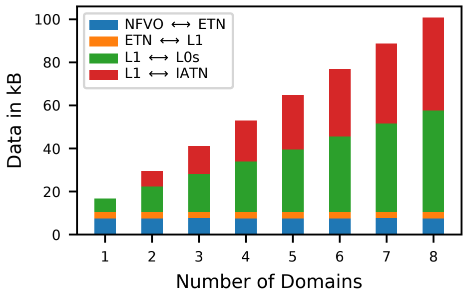
B. Signalling Overhead

Fig. 6(b) depicts the signalling overhead incurred when provisioning the *vl2-p2p* (left), *vl2-tree* (middle) and *vl2-broadcast* (right) services as the number of VNFs in the network service increases. The aggregate signalling is split in the following interfaces: i) *NFVO* ↔ *ETN*, ii) *ETN* ↔ *L1*, iii) *L1* ↔ *L0*, iv) *ETN* ↔ *Network*, measuring the OpenFlow traffic between the ETN Controller and the ETN data-plane elements, and v) *L0* ↔ *Network* measuring traffic between the L0 controllers and the Ethernet transport devices.

The aggregate signalling overhead exhibits the same growth behavior as the service provisioning time. However, the contribution of each interface varies across the three services. In the *vl2-p2p* service the interfaces contributing to the growth in signalling are the *NFVO* ↔ *ETN* and the *ETN* ↔ *Network* interfaces, as every time a new VNF is deployed the NFVO issues a new request and the remote ETN needs to be programmed. The worst case signalling for this service is below 65 kB. In the *vl2-tree* service, in addition to the previous two factors, we add the lineal growth of the *L1* ↔ *L0* interface, since a new LSP in the L0 domain is deployed for each new VNF, resulting in a worst case signalling below 300 kB. In the *vl2-broadcast* service, the quadratic increase is driven mostly by the *ETN* ↔ *Network* and the *L1* ↔ *L0* interfaces, since every new VNF pair requires a call to the L0 controller and programming new rules at the corresponding



(a) Service provisioning time



(b) Signalling overhead

Fig. 7. Results for Experiment 2

ETN, resulting in a worst case signalling below 4000 kB. It is worth noting that in all experiments the *L0* ↔ *Network* is low because we considered a simple L0 network topology in our emulation. Nevertheless, these results demonstrate that in a practical 5G networks vL2-WIM will introduce a small amount of signalling.

Fig. 7(b) depicts the signalling load when increasing number of L0 domains in experiment 2. The dominant factors are the *L1* ↔ *L0* as well as *L1* ↔ *IATN* interfaces, both increasing at similar linear rates. Signaling on the *L1* ↔ *IATN* interfaces only becomes necessary when more than one domain is involved in the VNF while each end-to-end LSP requires a call to all the involved L0 domains. Interactions required at the edge (ETNs) are constant. The worst case signalling load for this experiment with up to 8 domains is below 105 kB.

IV. CONCLUSION

Future 5G networks will be composed of distributed compute infrastructures that require novel data center interconnect technologies. In this paper, we propose vL2-WIM, a novel ETSI NFV WIM, which deploys virtualized L2 services across data centers. The main features of vL2-WIM are the ability to flexibly support customized virtual L2 service topologies, while minimizing signalling load across the WAN. Through a detailed experimental evaluation, we show that vL2-WIM can deploy virtual layer 2 service across 30 distributed compute locations in less than 15 seconds.

V. ACKNOWLEDGEMENTS

This work is supported by the European Commission's Horizon 2020 research and innovation program under grant agreement No 871428, 5G-CLARITY project.

REFERENCES

[1] Redana, Simone, et al. ”5G PPP Architecture Working Group: View on 5G Architecture.” (2019).

[2] Ersue, M., 2013, November. ETSI NFV management and orchestration-An overview. In Proc. of 88th IETF meeting.

[3] Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., and Folgueira, J. (2017). Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. IEEE Communications Magazine, 55(5), 80-87.

[4] ETSI, O. (2016). Open Source MANO: Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV.

[5] Slim, Farah, et al. ”Towards a dynamic adaptive placement of virtual network functions under ONAP.” 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE, 2017.

[6] Camps-Mur, D., Gutierrez, J., Grass, E., Tzanakaki, A., Flegkas, P., Choumas, K., ... and Legg, P. (2019). 5G-XHaul: A novel wireless-optical SDN transport network to support joint 5G backhaul and fronthaul services. IEEE Communications Magazine, 57(7), 99-105.

[7] Schmidt, Robert, Chia-Yu Chang, and Navid Nikaein. ”FlexVRAN: A flexible controller for virtualized RAN over heterogeneous deployments.” ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019.

[8] Juniper Networks, Inc. “VXLAN Data Center Interconnect Using EVPN Overview - TechLibrary - Juniper Networks.” https://www.juniper.net/documentation/en_US/junos/topics/concept/vxlan-evpn-integration-overview.html (Accessed Jun. 14 2020).

[9] Bartelt, J., Vucic, N., Camps-Mur, D., Garcia-Villegas, E., Demirkol, I., Fehske, A., ... and Lyberopoulos, G. (2017). 5G transport network requirements for the next generation fronthaul interface. EURASIP Journal on Wireless Communications and Networking, 2017(1), 1-12.

[10] Vilalta, R., López-de-Lerma, A. M., López, V., Mrówka, K., Szwedowski, R., Neidlinger, S., ... and Martínez, R. (2019, March). Transport API extensions for the interconnection of multiple NFV infrastructure points of presence. In Optical Fiber Communication Conference (pp. W1G-2). Optical Society of America.

[11] Simpson, Steven, Arsham Farshad, Paul McCherry, Abubakr Magzoub, William Fantom, Charalampos Rotsos, Nicholas Race, and David Hutchison. ”DataPlane Broker: Open WAN control for multi-site service orchestration.” (2019).

[12] Garrich, M., Hernández-Bastida, M., San-Nicolás-Martínez, C., Moreno-Muro, F. J., and Pavon-Marino, P. (2019, March). The Net2Plan-OpenStack Project: IT Resource Manager for Metropolitan SDN/NFV Ecosystems. In Optical Fiber Communication Conference (pp. M3Z-16). Optical Society of America.

[13] Giatsios, D., Choumas, K., Flegkas, P., Korakis, T., Cruelles, J. J. A., and Mur, D. C. (2019, May). Design and evaluation of a hierarchical SDN control plane for 5G transport networks. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[14] Muñoz, R., Mayoral, A., Vilalta, R., Casellas, R., Martínez, R., and Lopez, V. The need for a transport API in 5G networks: The control orchestration protocol. In 2016 Optical Fiber Communications Conference and Exhibition (OFC) (pp. 1-3). IEEE.

[15] Kourtis, Michail-Alexandros, et al. ”Enhancing VNF performance by exploiting SR-IOV and DPDK packet processing acceleration.” 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN). IEEE, 2015.

[16] Pfaff, Ben, et al. ”The design and implementation of open vswitch.” 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). 2015.

[17] Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., ... and Parulkar, G. (2014, August). ONOS: towards an open, distributed SDN OS. In Proceedings of the third workshop on Hot topics in software defined networking (pp. 1-6).

[18] L. Dunbar, W. Kumari, and I. Gashinsky, “Practices for Scaling ARP and Neighbor Discovery (ND) in Large Data Centers,” RFC 7342 (Informational), IETF, Aug. 2014. [Online]. Available: https://www.ietf.org/rfc/rfc7342.txt