

Experimental Validation of Compute and Network Resource Abstraction and Allocation Mechanisms within an NFV Infrastructure

Ricardo Martínez, Luca Vettori, Jorge Baranda, Josep Mangues and Engin Zeydan
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels, Spain
ricardo.martinez@cttc.es

Abstract—5G supported capabilities (e.g., *slicing*) enable accommodating heterogeneous vertical services having their own requirements over a common cloud and transport infrastructure. In this context, the EU-H2020 5Growth project defines a service and infrastructure orchestration architecture to automatically deploy network services (NSes) fulfilling vertical demands. In this architecture, the Service Orchestrator (5Gr-SO), as a *service provider*, maps the vertical service needs into NS requirements (e.g., CPU, RAM, bandwidth, etc.). The 5Gr-SO interacts with an underlying infrastructure orchestrator referred to as 5Gr-RL. The 5Gr-RL, as an *infrastructure provider*, handles two main functions: i) *abstraction* of the resources exposed to the 5Gr-SO, and ii) *fine-grained resource selection*. Different interaction forms between both 5Gr-SO and 5Gr-RL arise differing in the exchanged abstracted information and resource allocation. We present two 5Gr-SO and 5Gr-RL interaction solutions stemming from two 5Gr-RL operational modes: Infrastructure Abstraction (InA) and Connectivity Service Abstraction (CSA). In the InA approach, the 5Gr-SO is granted with an aggregated view of the computing resources and a set of transport *logical links* between the cloud locations. In the CSA strategy, besides the aggregated view of the cloud resources, the *logical links* are associated to potential connectivity service types. Both InA and CSA strategies are presented describing their pros and cons. Moreover, the designed workflows (involving the devised abstraction and allocation algorithms) between the 5Gr-SO and 5Gr-RL entities are experimentally validated. Scalability studies are conducted upon two different cloud and transport infrastructure sizes in terms of the abstraction composition time, the expansion computation time, and total NS deployment time.

Index Terms—Network Services, Resource Abstraction

I. INTRODUCTION

The 5G networking community is designing architectural constructs to offer heterogeneous multi-technology and multi-domain network and computing infrastructure to accommodate the requirements of diverse vertical industries [1]. Vertical services are translated into low-level networking and computing requirements as Network Services (NSes), which are eventually rolled out over a common infrastructure (Network Function Virtualization Infrastructure, NFVI). Typically, the

NFVI is made up of distributed cloud locations (i.e., NFVI-Point-of-Presence, NfviPops) interconnected by a transport network (Wide Area Network, WAN). In this context, the EU-H2020 5Growth project defines an end-to-end service and resource orchestration stack [3] where NSes (defined by Virtual Network Functions, VNFs and virtual links, VLs) are automatically deployed over the NFVI. Fig. 1 depicts the key building blocks of the 5Growth architecture. The Service Orchestrator (5Gr-SO) operates as the *service provider*. It receives the vertical (NS) requests from the Vertical Slicer (5Gr-VS) and selects the NFVI resources fulfilling the NS requirements. To do so, the 5Gr-SO is implemented on top of the open source management and orchestration (MANO) [2] along with interacting with a lower-layer infrastructure orchestrator (i.e., Resource Layer, 5Gr-RL). The 5Gr-RL acts as the *infrastructure provider* to provide required NFVI abstraction functionality to 5Gr-SO. This NFVI abstraction entails an aggregated view of the computing resources (i.e., CPU, RAM and Storage) as well as the logical links (LLs) supporting the NfviPops' connectivity. To do so, the 5Gr-RL coordinates a pool of dedicated controllers such as the Virtualized Infrastructure Manager (VIMs) (e.g., OpenStack) and the WAN Infrastructure Manager (WIM) to perform the resource selection and allocation [4].

In this work, we focus on the orchestration functions executed when processing a new NS request that specifies the compute and networking requirements needed by a vertical service. Those functions entail the interactions and operations between the 5Gr-SO and the 5Gr-RL to conduct the required NFVI abstraction and resource selection/allocation. For the sake of clarity, the NFVI abstraction is defined as the 5Gr-RL functionality exposing an aggregated view of the (computing and networking) resources to the 5Gr-SO. On the other hand, the resource selection/allocation embraces those mechanisms (performed at both 5Gr-SO and 5Gr-RL) to map the NS requirements into the available NFVI view. Such a mapping can be devised to attain e.g., an efficient use of the overall NFVI resources. The design of both NFVI abstraction and resource selection becomes essential since they state a trade-off between the abstraction level accuracy and the optimality of the resource selection [5].

In light of the above, we propose two 5Gr-RL operational

Work supported in part by EU Commission H2020 5Growth project (Grant No. 856709), Spanish MICINN AURORAS (RTI2018-099178-B-I00) and Spanish MINECO 5G-REFINE (TEC2017-88373-R) projects and Generalitat de Catalunya grant 2017 SGR 1195.

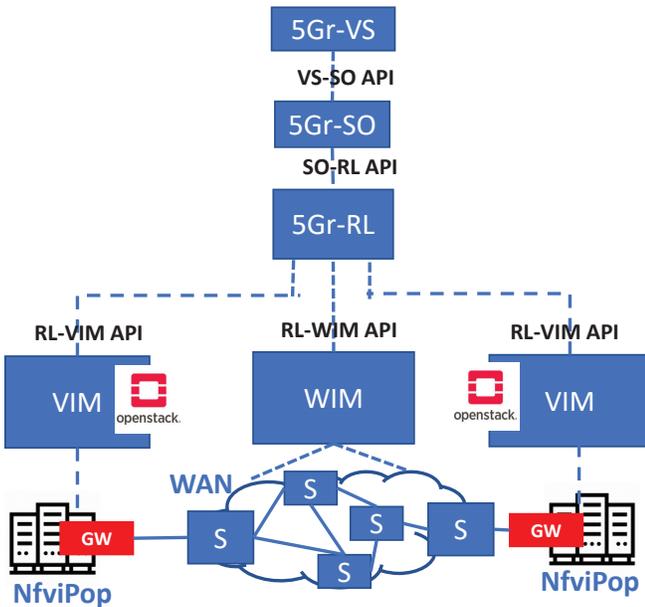


Fig. 1: 5Growth Stack

modes called Infrastructure Abstraction (InA) and Connectivity Service Abstraction (CSA). Both modes differ on the adopted NFVI abstraction strategy to derive the LLs between NfviPops and the subsequent mapping of the required NS resources. In the InA approach, upon 5Gr-SO request, the exposed LLs by the 5Gr-RL are associated to explicitly feasible WAN paths between each NfviPop pair. This allows the 5Gr-SO selecting the NfviPops (hosting the NS's VNFs) and the corresponding LLs. This approach is interesting for scenarios where both the *service orchestrator* (i.e., 5Gr-SO) and the *resource orchestrator* (i.e., 5Gr-RL) are managed by a single NFVI operator. On the other hand, in the CSA model, the set of LLs are bound to a pre-determined set of connectivity service types/classes (e.g., *Gold*, *Silver* and *Bronze*) offered by the 5Gr-RL. Each connectivity service type defines their own features (e.g., minimum guaranteed bandwidth, maximum tolerated delay, etc.). Once the 5Gr-SO selects the NfviPops and LLs, the 5Gr-RL needs to conduct specific WAN path computations (i.e., expansion function) to fulfill the service type parameters associated to every LL. The CSA mode may become a plausible solution for scenarios where both service and resource orchestrators are managed by different operators (e.g., to avoid sharing NFVI details for confidentiality reasons).

The resulting interactions (i.e., workflows) and devised abstraction and expansion algorithms for both InA and CSA approaches are presented and experimentally validated. This is used to conduct scalability studies comparing the performance attained by both modes with respect to the abstraction and resource allocation time and total NS deployment time. This also draws the pros and cons brought by each operational mode. To this end, two different NFVI with different emulated WAN sizes (i.e., a Core WAN and a Small Metro WAN) are

considered.

The paper is organized as follows: in Sec. II we review some of the works produced in the literature addressing the resource abstraction and allocation. In Sec. III, it is discussed the 5Growth resource views at both 5Gr-SO and 5Gr-RL. Next, Sec. IV details the proposed 5Gr-RL InA and CSA operational modes detailing their workflows and associated abstraction and resource allocation (RA) algorithms. The experimental validation of InA and CSA strategies along with the scalability discussion on selected figures of merit are tackled in Sec. V. Finally, in Sec. VI conclusions and future work are drawn.

II. RELATED WORK

The orchestration and automatic deployment of end-to-end 5G NSes/slices (involving both computing and transport resources) over a common multi-domain and multi-technology NFVI has received notable attention in the last years. Typically, two orchestration and control layers are involved: the *service orchestrator* and the *infrastructure orchestrator* [6], [7]. The former deals with the service provider (herein called as the 5Gr-SO) and takes over of the end-to-end service creation. This involves the mapping of the NS's VNFs and VLs over an abstracted NFVI view provided by the underlying infrastructure orchestrator, referred in this work as the 5Gr-RL. Focusing in the abstraction of the transport network, the aim is to present to the service orchestrator a virtual representation of the network domain/s for the remote NfviPop connectivity. In this context, the authors in [5] present different abstraction models. One of them is the *big switch* strategy where the transport network domain is modelled as a single node (switch) only identifying the endpoints attached to the NfviPops. In other words, the service orchestrator operates with no LLs reflecting the remote NfviPop connectivity. Thus, the service orchestrator is constrained to select the NfviPops for the NS' VNFs and delegates the eventual inter-NfviPop connectivity to the lower-layer infrastructure orchestrator. This approach presents some limitations since it is likely that the selected NfviPops for the NS cannot be successfully completed due to lack of a feasible transport connection (e.g., either bandwidth or latency requirements cannot be fulfilled). Another approach proposed in [5], [7], [8], relies on advertising a set of point-to-point LLs to the service orchestrator. These LLs are indeed supported by end-to-end connections computed by the infrastructure orchestrator. The exposed LLs inherit the attributes (e.g., available bandwidth, delay, etc.) of the associated transport network connections. Thereby, the service orchestrator can make a more efficient selection of the NfviPops being aware of the LLs' features. Finally, in [9] and [10], both works exploit the idea to advertise the LLs as defined connectivity services offered by the infrastructure orchestrator. Consequently, the LLs are not subject to a explicit end-to-end transport network connection. Rather the LLs are bound to a defined set of QoS classes where each class specifies attributes such as elastic bandwidth (i.e, maximum and minimum transport capacity), permitted packet loss rate, maximum permitted delay, availability, etc.

Once the service orchestrator selects the LLs, the infrastructure orchestrator must find the transport resources to satisfy the LLs class attributes. In this paper, some of these concepts and ideas addressed in the previous works are leveraged to design and implement the targeted InA and CSA operational modes within the 5Growth architecture.

III. 5GROWTH NFVI VIEW: ABSTRACTION AND EXPANSION FUNCTIONS

Fig. 2 depicts the adopted NFVI view handled by both the 5Gr-SO and 5Gr-RL entities. It is worth recalling that the NFVI embraces two resource types: i) those related to the computing resources (i.e., CPU, RAM and Storage) within an NfviPop; ii) those related to the transport network (WAN) resources for the NfviPops inter-connections. For the computing resources, the 5Gr-RL (interacting with the VIM controllers) retrieves (in an aggregated way) the total and available (i.e., not used) amount of CPU, RAM and Storage at every NfviPop. The same computing resource view is also kept by the 5Gr-SO. However, for the transport network resources, the view at both elements differ. Specifically, the 5Gr-RL interacts with the WIM controller/s to create the WAN network topology (nodes and link connectivity). For each link, the 5Gr-RL is aware of a set of attributes characterizing the network link such as the available bandwidth (in b/s), the associated delay (in ms), the cost, etc. [11], [12].

As said, the abstracted WAN view at the 5Gr-SO is formed by the LLs as shown in Fig. 2. Indeed, the LLs constitutes a virtual representation of the (potential) physical connectivity between a pair of NfviPops. Every LL has its own attributes in terms of bandwidth, delay, cost, etc. The adopted 5Gr-RL operational mode (either InA or CSA) determines how this LLs are derived and then exposed to 5Gr-SO. In the InA mode, every LL is always associated to a feasible computed WAN path done at the 5Gr-RL. As an example, the LL connecting NfviPop1 and NfviPop2 in Fig. 2 is derived from the computed path formed by NfviPop1, S1, S5, S4, S2 and NfviPop2. The description of the InA abstraction algorithm is addressed in Sec. IV-A. The LL attributes inherit the features of the underlying WAN path supporting such an LL. That is, the available LL bandwidth is inferred from the unused bandwidth on the most congested path link. Moreover, the LL delay is computed as the total accumulated delay over each WAN path link. Observe, that once the 5Gr-SO selects a specific LL for the NS, it implicitly determines the WAN nodes and links to accommodate the traffic flow between the respective NfviPops.

In the CSA mode, the LLs are not bound to a pre-computed WAN path. Indeed, the 5Gr-RL defines a set of service types/classes which define the supported characteristics of the candidate NfviPop pair connectivity. Therefore, no explicit abstraction computation is triggered to derive the exposed LLs to the 5Gr-SO. To illustrate that, let's assume that three connectivity service types (i.e., Gold, Silver and Bronze) are offered by the 5Gr-RL. Each type/class provides its own parameters; guaranteed bandwidth and maximum delay. Following the above example, the 5Gr-RL offers to the 5Gr-SO

two LLs for the NfviPop1 and NfviPop2 connectivity based on both Silver and Bronze service types. Conversely to the InA mode, once the 5Gr-SO selects a LL for the NS, the 5Gr-RL is then required to conduct an explicit WAN path computation (i.e., expansion function) to find the nodes and links supporting the selected LL attributes, i.e., the associated service type/class parameters.

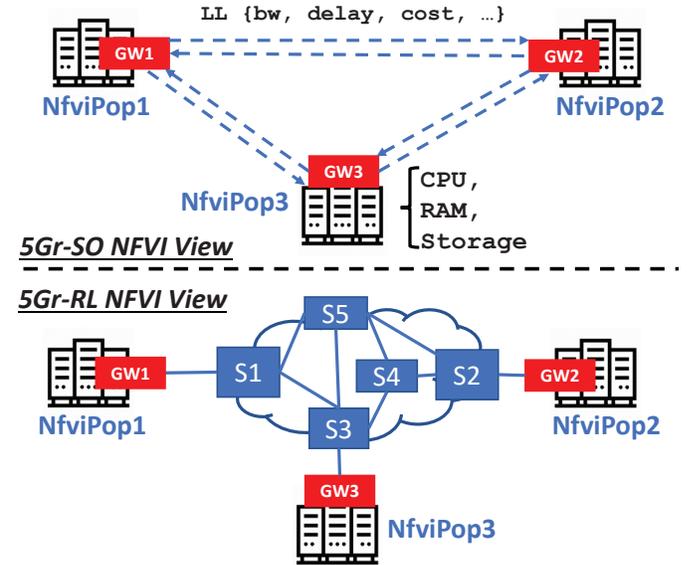


Fig. 2: 5Growth stack resource view

The abstraction and expansion computations for the InA and CSA operational modes are carried out within the 5Gr-RL, and are referred to as Resource Allocation (RA) algorithms. Both RA algorithms are implemented as an standalone process called 5Gr-RL RA Server¹ and detailed next.

IV. RESOURCE ALLOCATION (RA) ALGORITHMS

The RA algorithms for the InA and CSA modes are presented in this section. Prior to that, for each mode, it is presented the subjacent interactions and operations (i.e., workflows) between the 5Gr-SO and 5Gr-RL entities upon processing a new NS request (*NSReq*). Indeed, the respective InA and CSA mode workflows define the design and objective functions pursued by their associated RA algorithms.

A. RA Infrastructure Abstraction (InA) Algorithm

The workflow implementing the InA operational mode is depicted in Fig. 3 and start once the 5Gr-SO receives a new *NSReq*. An *NSReq* specifies an NS descriptor detailing the set of VNFs (with the required amount of CPU, RAM and Storage) as well as the VNs to be deployed among the VNFs. Typically, the VNs specifies network-oriented requirements, e.g., bandwidth (in b/s) and maximum latency (in ms).

To serve an incoming *NSReq*, the 5Gr-SO first asks (step 1a) to the 5Gr-RL the NFVI abstraction view (i.e., NfviPops and LLs resources). This is done via a RESTful API. For the

¹https://github.com/5growth/5gr-rl/tree/master/rl/RL_RA_Server_R1/src

NfviPop resources, the 5Gr-RL communicates (step 2) with every NfviPop controller (VIM) to retrieve the aggregated amount of CPU, RAM and Storage. Similarly, the 5Gr-RL requests to the WIM information about transport WAN, i.e., nodes and links' attributes (step 3). The received WAN information is then used as input parameter to query the RA server (step 4a) for computing (up to K) WAN paths to interconnect every possible NfviPop pair within the NFVI. The WAN path computations are done by the devised InA algorithm at the RA server (step 4b), which detailed below. The resulting (K) WAN paths for each NfviPop pair derives the set of LLs (and their attributes) being exposed to the 5Gr-SO (step 1b).

Fig. 4 shows the JSON contents carried in the HTTP response message made in step 1b. These contents describe the abstracted view of both sets: NfviPops and LLs. For each NfviPop, it is detailed the information such as geographical location aspects, network connectivity endpoint (i.e., IP address of the NfviPop Gw), NfviPop identifier, and the VIM identifier managing that specific NfviPop. Additionally, it is also determined (in an aggregated way) the NfviPop total and available compute resources in terms of CPU, memory and storage. This abstract model is related to the information model described in [16]. On the other hand, for each LL, the exposed abstract information (based on [17]) is: the IP addresses of the endpoints (i.e., NfviPop's Gws) inter-connected by the LL, the LL identifier, the local and remote identifiers (required if multiple LLs inter-connect a given NfviPop pair), the associated QoS attributes (i.e., cost and delay), and the available bandwidth.

After receiving the abstracted NFVI view, the 5Gr-SO triggers (step 5) a placement algorithm (out of the scope of this work) to choose the NfviPops (hosting the VNFs) along with the LLs enabling the NfviPop connectivity. The latter are used to accommodate the VLs between VNFs at different NfviPops. In step 6, the 5Gr-SO (using OSM MANO) coordinates the VNF deployment at the selected NfviPops through communicating with the VIMs. Details of these interactions (based on OpenStack API) are explicitly omitted for the sake of space. Next, the selected LLs are notified (step 7a) to the 5Gr-RL. In this request, it is specified the amount of bandwidth to be allocated on each LL. Note that this requested bandwidth is associated to the requirements of the VLs. For each selected LL, the 5Gr-RL picks its previously computed WAN path (done in step 4b). The workflow is completed in step 8, where the flows on the involved WAN paths (i.e., nodes and links) are programmed by the corresponding WIM controller.

In step 4b of the workflow, the RA InA algorithm computes the WAN paths used to derive the LLs. The inputs of this algorithm are: i) the request set ($reqSet$) describing all the NfviPop pairs; ii) the maximum number (K) of WAN paths to be computed for each request (r_x) in $reqSet$; and iii) the WAN view (G). The rationale behind targeting k WAN paths deriving K LLs for an NfviPop pair is to enable the 5Gr-SO to use different LLs if multiple VLs need to be deployed between a given NfviPop pair. The output of algorithm is formed by a path set ($pSet$) containing the K WAN paths bound to each

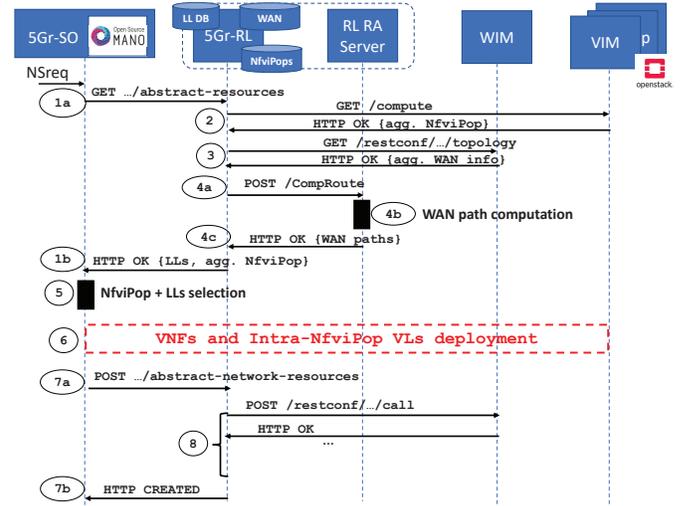


Fig. 3: Workflow in the 5Growth stack for RA InA

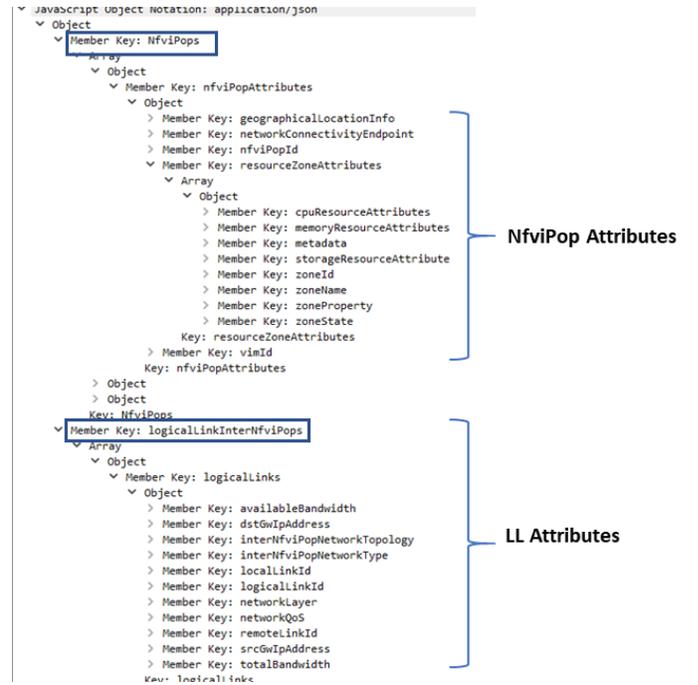


Fig. 4: JSON encoding for abstracted NfviPops and LLs

r_x . The following notation is used to discuss the devised and implemented RA InA algorithm described in Algorithm 1:

$reqSet$

Requests with all possible ingress ($NfviPop_i$) and egress ($NfviPop_j$) pairs, $i \neq j$. An NFVI with N NfviPops entails a $reqSet$ with $N * N - 1$ r_x ;

r_x

$r_x \in reqSet$ for $x : 1..(N * N - 1)$;

r_x^i, r_x^j

$NfviPop_i$ and $NfviPop_j$ of r_x ;

r_x^b, r_x^l

Bandwidth and latency for r_x ;

G

WAN graph, i.e, nodes and links attributes;

p_{r_x}

K WAN paths between for r_x ;

$p_{r_x}^w$

$p_{r_x}^w \in p_{r_x}$ for $w : 1..K$;

$p_{r_x}^w.b$	Available bandwidth for $p_{r_x}^w$;
$p_{r_x}^w.l$	Latency for $p_{r_x}^w$;
$p_{r_x}^w.cb$	Path cost for $p_{r_x}^w$;
$f_c(r_x, G, K)$	Compute K WAN paths for r_x , i.e., connecting r_x^i and r_x^j fulfilling r_x^b and r_x^l ;
$sort(p_{r_x})$	Sort $p_{r_x}^w$ according to $p_{r_x}^w.b, p_{r_x}^w.l, p_{r_x}^w.c$ for $w : 1..K$;
$pSet$	Set with computed p_{r_x} for $x : 1..(N*N-1)$;

For each r_x , the K shortest paths (K-SP) relying on the well-known Yen algorithm [14] are computed (i.e., f_c function). The targeted objective function of the algorithm is to compute the *best* K WAN paths for r_x . This is done seeking for the WAN paths having the largest end-to-end available bandwidth, shortest delay and lowest cost. Thus, no requirements in terms of bandwidth and end-to-end latency are passed to the algorithm. The computed paths (p_{r_x}) are sorted (via the $sort()$ function) according to: 1) paths with largest available bandwidth (i.e., $p_{r_x}^w.b$); 2) paths with lowest end-to-end latency (i.e., $p_{r_x}^w.l$); 3) paths with lowest cost (i.e., $p_{r_x}^w.c$). Herein the cost is determined by the number of hops. The computed p_{r_x} are appended to the output $pSet$. Once all the r_x are processed, the resulting $pSet$ is used by the 5Gr-RL to create the associated LLs.

In light of the above, an offered advantage bound to the InA operational mode is that the placement mechanism at the 5Gr-SO operates with a set of LLs whose underlying WAN paths are feasible. This may favour the 5Gr-SO LL selection to better fulfill the NS requirements. This is attained at the expenses of querying the 5Gr-RL computing all the WAN paths to derive the LLs; depending on the WAN size, this may cause scalability issues in terms of high computation burden, increased NS deployment times, etc.

Algorithm 1 RA InA pseudocode

Input: $reqSet, K, G$

Output: $pSet$

- 1: **for** r_x in $reqSet$ **do**
 - 2: $p_{r_x} \leftarrow f_c \text{comp}(r_x, G, K)$ {K}-SP computation
 - 3: $p_{r_x} \leftarrow sort(p_{r_x})$
 - 4: $pSet \leftarrow pSet + p_{r_x}$
 - 5: **end for**
 - 6: **return** $pSet$
-

B. RA Connectivity Service Abstraction (CSA) Algorithm

The basic idea behind the CSA operational mode is that the exposed LLs are related to connectivity service types offered over the WAN. This makes that an abstraction mechanism is not triggered at the 5Gr-RL to infer the LLs. That said, once the 5Gr-SO selects the LLs to accommodate a $NSreq$, the 5Gr-RL needs to seek for the WAN paths that fulfil those LL requirements. This path computation made by the 5Gr-RL is what herein is defined as the expansion mechanism.

Fig. 5 depicts the workflow for the CSA operational mode. In step 1a, the 5Gr-SO requests the NFVI abstraction to the 5Gr-RL. NfviPop resources are directly asked to the VIMs'

NfviPops as shown in step 2. As for the LLs, the 5Gr-RL advertises its pre-defined LL database mapping the LLs with the supported connectivity service types. Both NfviPop and LLs are exposed to the 5Gr-SO in step 1b, which then executes the placement mechanism (step 3). The $NSreq$'s VNFs are deployed at the selected NfviPops (step 4). After that, the 5Gr-SO commands the 5Gr-RL allocating the WAN resources to fulfil the selected LLs' needs. For each of these LLs, the steps 6 – 8 are sequentially repeated. First, in step 6 the 5Gr-RL updates its WAN view. After this, the RA CSA algorithm is queried to find the WAN path fulfilling the LL requirements (steps 7a). This request contains the LL's ingress and egress NfviPops, the bandwidth and latency requirements, along with the retrieved WAN view. The RA CSA algorithm is executed (step 7b) yielding the LL expansion over a feasible WAN path. Last but not least, the RA CSA output (WAN path) is processed by the 5Gr-RL to eventually conduct the flow programmability over the path (step 8).

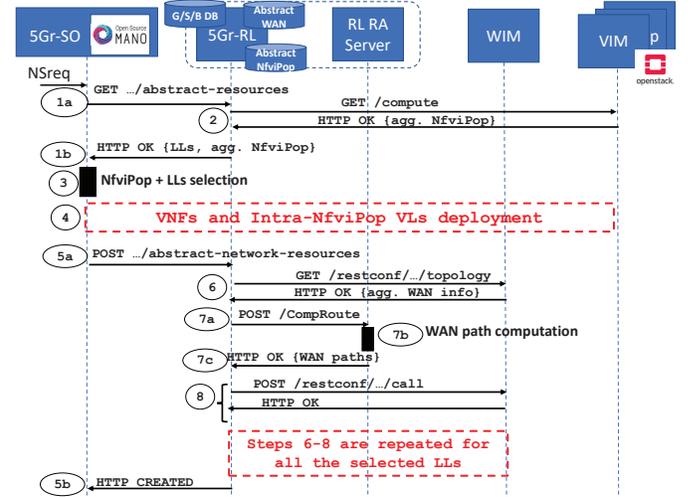


Fig. 5: Workflow in the 5Growth stack for RA CSA

Algorithm 2 describes the pseudocode of the RA CSA mechanism. The input information is formed by: i) the request r_x ($x = 1$) specifying the ingress/egress NfviPop pair for the targeted LL; ii) the required bandwidth (r_x^b) and the maximum latency (r_x^l); and iii) the updated WAN information (G). The output of the RA CSA algorithm specifies the expanded WAN path (p_{r_x}) fulfilling the bandwidth and latency requirements.

Algorithm 2 RA CSA pseudocode

Input: r_x, G

Output: p_{r_x}

- 1: $p_{r_x} \leftarrow f_c(r_x, K, G)$
 - 2: **return** p_{r_x}
-

For the received r_x , the K-SP computation is done using the f_c . For each $p_{r_x}^w$ path, the bandwidth (r_x^b) and latency (r_x^l) requirements must be dealt with. The selected path (p_{r_x}) out of the computed K paths is the one having the highest

available bandwidth. If two or more paths present the same unused bandwidth, the tie is broken choosing the one with the lowest end-to-end latency. If the tie persists, the lowest path cost is picked. The pr_x is then returned to the 5Gr-RL to complete the flow path node configuration (via the WIM).

The CSA operational mode does reduce the computational burden at the 5Gr-RL to infer the LLs (i.e., the abstraction mechanism). This, however, has two implications: i) the selected LLs made by the 5Gr-SO when serving a *NSReq* may not be eventually allocated by the 5Gr-RL. This happens if the expansion function done by the CSA algorithm is unable to find a feasible path satisfying the LL requirements; ii) the CSA expansion algorithm is executed for each individual LL chosen by the 5Gr-SO. To do so, the 5Gr-RL must interact with the WIM (to update the WAN view) before performing each particular path computation. The latter does increase notably the 5Gr-RL - WIM interactions.

V. EXPERIMENTAL VALIDATION

This section tackles the experimental validation of the 5Gr-RL operational modes, i.e., devised workflows and RA algorithms. Two separated NFVI scenarios differing in the WAN size (i.e., number of nodes and links) are considered. The aim is to conduct a scalability evaluation of both CSA and InA modes under such diverse WAN infrastructures. To this end, several metrics are used: i) number of 5Gr-RL and RL RA Server interactions; ii) the *RA computation time* defined as the elapsed time between the request message is sent by the 5Gr-RL and the response message is returned by the RL RA server; iii) the *total number of abstracted LLs*; iv) the *abstracted information time* determined as the elapsed time between the 5Gr-SO sends the request and receives the response with the abstracted NFVI; and v) the *LL deployment time* entailing the total time to deploy the WAN paths supporting the selected 5Gr-SO selected LLs.

A. Considered WAN Scenarios and Main Assumptions

Fig. 6 depicts the two considered NFVIs. Each one is formed by a single domain with three NfviPops. The WAN interconnecting the NfviPops is either: a) a core WAN (designed to mimic a Spanish national WAN) formed by 14 packet-switch nodes and 22 bidirectional links (Fig. 6(a)); b) a *small* metro network with 5 packet-switch nodes and 8 bidirectional links (Fig. 6(b)). Every link has a transport capacity of 1 Gb/s where the associated delay (in ms) is labelled on the link. The connectivity between the NfviPop termination point (i.e., Gw) and attached packet switch (e.g., NfviPop1 and S1 in Fig. 6(b)) provides a net data rate of 1 Gb/s with negligible link delay.

In both NFVIs scenarios, a centralized WIM (i.e., SDN controller) takes over of the WAN programmability with emulated flow configuration. Received *NSReqs* are handled by the 5Gr-SO (via an Open Source MANO (OSM) implementation). The 5Gr-SO is based on the CTTC open source Multi-nfvo federated ServiCe orchestrAtion pLatform (MESCAL)² [15].

Three different VIMs (OpenStack) manage the respective NfviPops. The VIMs and WIM instances are coordinated by the 5Gr-RL to de-/allocate the NFVI resources. The 5Gr-RL relies on the CTTC open source implementation called *mobile transport pLatform for multi-tEchnology neTwoRk infrAstructure* (ELECTRA)³ [4]. The RL RA Server, where both RA InA and CSA algorithms are hosted, is also an open source implementation⁴. The 5Growth stack components (i.e., 5Gr-SO, 5Gr-RL, RL RA Server, VIMs and WIM) are installed over different Linux-based servers at the CTTC 5G end-to-end platform [15].

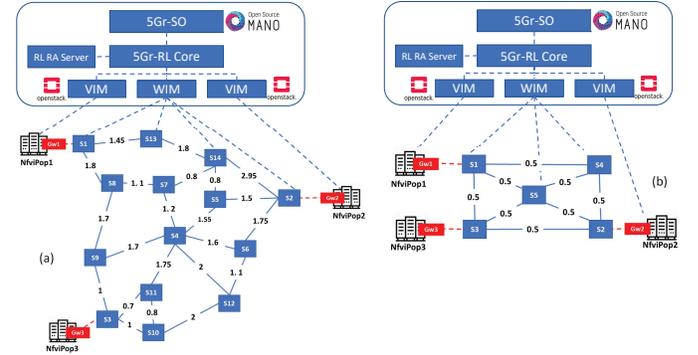


Fig. 6: (a) core WAN; (b) *Small* metro WAN

To conduct the validation and scalability evaluation, we rely on an *NSReq* (Fig. 7) based on the eHealth use case investigated in the completed EU 5G-Transformer project [13]. Such *NSReq* requires a NS descriptor formed by six VNFs: 5 VNFs (i.e., MME, PGW, HSS, SGW and SEC-GW) which do implement the entities of a virtualized evolved packet core (vEPC); the remaining VNF is a server VNF working as a back-end entity for a specific eHealth application. Moreover, the targeted *NSReq* encompasses six VLs supporting the different interfaces present in the EPC system, i.e., SGi, S5, S8, S6b, S1 and a VL shared between all the VNFs for the management traffic. The VLs are shown in Fig. 7 with solid lines, whilst the numbers represent the NS service access points. In this paper, we are not interested in the internals of the application, but only on its complex NS topology. It is assumed that the 5Gr-SO placement function distributes the set of VNFs among the three available NfviPops. Specifically, the SEC-GW and SGW VNFs are deployed at NfviPop1, the rest of vEPC entities (i.e., MME, PGW and HSS) are allocated at NfviPop2, and the server VNF is placed at NfviPop3. As a result of this VNF distribution, the 5Gr-SO requires to to the 5Gr-RL 15 inter-NfviPop connections (i.e., LLs).

Exclusively for the CSA operational mode and without loss of generality, table in Fig. 8 details the offered LLs for both core and Small WANs considering three defined Gold/Silver/Bronze connectivity service types/classes. Each service type defines a pair of guaranteed parameters: maximum bandwidth and delay. Observe that depending on the

²<https://github.com/5growth/5gr-so>

³<https://github.com/5growth/5gr-rl/tree/cttc-rl>

⁴https://github.com/5growth/5gr-rl/tree/master/rl/RL_RA_Server_R1

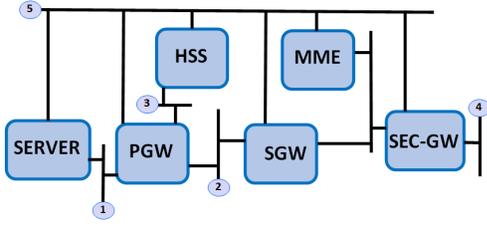


Fig. 7: Structure of the Monitoring NS under evaluation

WAN size, high performance connectivity service types (e.g., Gold) could not be offered because the maximum permitted latency/delay between a pair of NfviPops is exceeded. In the considered WAN scenarios, for the Small one, all three defined connectivity service types are allowed for any NfviPop pair. Nevertheless, in the core WAN, this does not occur.

WAN Scenario	NfviPop connectivity	Supported Connectivity Service Types	Connectivity Service Type Features [Max. Bw, Max Latency]
Spain WAN	NfviPop1 <-> NfviPop2	Bronze	Gold: [100Mb/s, 3ms] Silver: [50Mb/s, 5ms] Bronze: [25Mb/s, 10ms]
	NfviPop1 <-> NfviPop3	Silver & Bronze	
	NfviPop2 <-> NfviPop3	Bronze	
Small WAN	NfviPop1 <-> NfviPop2	Gold, Silver & Bronze	
	NfviPop1 <-> NfviPop3	Gold, Silver & Bronze	
	NfviPop2 <-> NfviPop3	Gold, Silver & Bronze	

Fig. 8: Supported Connectivity Service Types and LLs for the considered WAN scenarios in the RA CSA approach

B. Captured RESTful-based Control Messages

Figure 9 shows the captured control messages among the 5Growth stack (i.e., 5Gr-SO, 5Gr-RL, RA Server, VIM and WIM) when deploying a *NSReq*. Specifically, Fig. 9a describes the exchanged messages supporting the 5Gr-RL InA strategy workflow discussed in see Sec. IV-A. Similarly, Fig. 9b shows the control messages according to the workflow followed by 5Gr-RL CSA approach tackled in Sec. IV-B). For the sake of clarity, all the message captures are done at the same server device hosting the 5Gr-RL element. The captured messages as they are showed were edited to show the relevant operations aligned with the presented InA and CSA workflows.

C. Results and Discussion

Table I shows the numerical results obtained when adopting either the InA or CSA operational modes for both the core and Small WAN topologies when deploying the described *NSReq*. These results are given in terms of: i) number of RL RA algorithm executions (either for abstraction or expansions functions); ii) the elapsed RA InA and CSA algorithm computation time; iii) the number of abstracted LLs exposed by each operational mode to the 5Gr-SO; iv) the abstraction information time; and v) the time required by the 5Gr-RL (in both operations modes) to successfully deploying a received LL over the WAN path⁵. The goal is to evaluate and

⁵Since both WAN topologies are emulated, the represented time accounts for the time to make the expansion operation without considering the actual configuration of the path forwarding elements.

compare the scalability (using the two considered WAN size infrastructures) attained by InA and CSA operational modes.

According to the assumed VNF distribution through the three available NfviPops (detailed above) 15 inter-NfviPops connections (LLs at the 5Gr-SO context) are required to be deployed. In the 5Gr-RL, these inter-NfviPop connections are then rolled out over WAN paths. It is worth highlighting that a selected LL by the 5Gr-SO may accommodate more than one inter-NfviPop connection. In the InA operational mode, this means that such inter-NfviPop connections are set up over the same WAN path. Nevertheless, this may not happen in the CSA mode. That is, since the CSA mode triggers independent expansion WAN path computations for each inter-NfviPop connection, it is likely that even if two (or more) inter-NfviPop connections are bound to the same LL (e.g., Gold LL between a given NfviPop pair), the resulting WAN paths are different.

In the InA operational mode, the abstraction algorithm is executed once to come out with LLs exposed to the 5Gr-SO. As expected, the WAN size notably impacts on the RA InA abstraction computational time: 290 ms and 77 ms for the core and Small WAN topologies, respectively. Thus, the larger the WAN topology is, the more connectivity options between NfviPop pairs can be explored. This leads to increase the abstraction computation time. On the other hand, the RA algorithm for the CSA operational mode is exclusively triggered for the expansion WAN path computation. For the targeted *NSReq* requiring 15 inter-NfviPop connections, the RA CSA expansion algorithm is triggered 15 times. For each of them, it is needed to: i) retrieve the WAN information, and ii) execute the RA algorithm for the selected LL supporting an inter-NfviPop connection. Thereby, the RA CSA algorithm time also depends on the WAN size. For the core WAN, this computation takes around 22 ms, whilst for the Small WAN it requires 14 ms.

The number of LLs advertised to the 5Gr-SO varies according to the adopted 5Gr-RL operational mode. In the InA approach, for both core and Small WAN topologies, the number of abstracted LLs are 18. This is obtained considering $K = 3$ WAN paths resulting from inferring 3 (unidirectional) LLs for every NfviPop pair. For the RA CSA approach, the number of abstracted LLs depends on the feasibility to support the defined connectivity service types for each NfviPop pair. As depicted in the table within Fig. 8, the CSA approach for the core WAN scenario limits the number of exposed LLs to 8. For this topology and the defined connectivity service types, there are some NfviPop pairs where Gold and/or Silver service types cannot be offered since no WAN path fulfils the maximum end-to-end delay for those service types. Conversely, for the Small WAN topology, all the connectivity service types are offered for every NfviPop pair, resulting in 18 LLs exposed to the 5Gr-SO.

The abstracted information time is also a relevant metric. Regardless of the WAN scenario, the CSA approach attains lower time than in the InA mode. This is due to the fact that in the CSA mode no explicit abstraction path computation to derive the LLs is triggered. Focusing on the InA operational

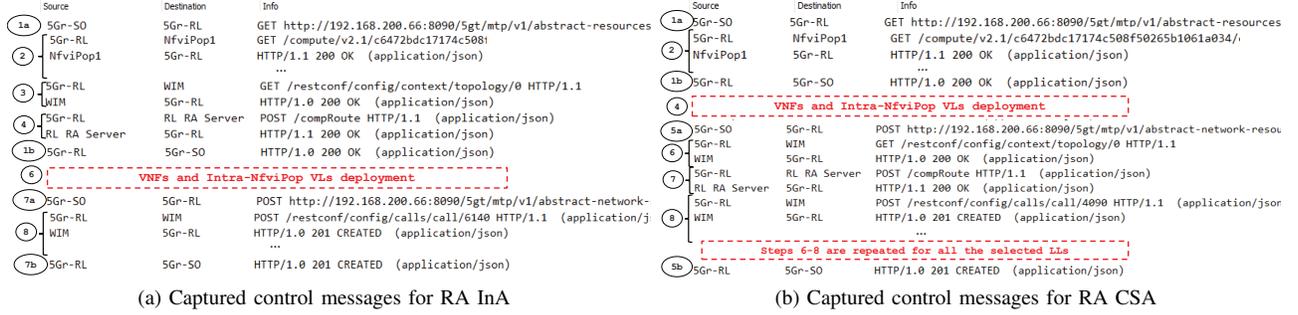


Fig. 9: Validation of the workflows of InA and CSA RAs within core WAN topology.

mode, again larger WAN infrastructures does increase the abstraction RA computation time that does enlarge the total abstraction information time.

Finally, the attained LL deployment time, i.e., deploying an inter-NfviPop connection over a feasible WAN path, results in more time consuming in the CSA mechanism when compared to the InA approach. This behaviour is observed in both considered WAN topologies (e.g., in the core WAN, the LL deployment achieved by the CSA mode takes 4708 ms, whilst in the InA this time is 2595 ms). Indeed, as discussed above, in the CSA approach the deployment of every inter-NfviPop connection over a selected LL entails: i) retrieving an updated view of the WAN infrastructure; ii) triggering the expansion WAN path computation for the inter-NfviPop connection; iii) conducting the packet flow configuration over the computed WAN path. However, in the InA approach, each inter-NfviPop connection to be rolled out over a selected LL has already determined the WAN path. Thus, there is not need to neither retrieving the WAN status nor executing a WAN path computation. This explains the reason why the CSA approach requires larger time when deploying an inter-NfviPop connection over a selected LL than in the InA approach.

In light of the above results, both InA and CSA operational modes has their own pros and cons. The InA approach requires a more complex and thus time consuming abstraction mechanism for deriving the LLs than the CSA approach. On the other hand, the deployment of the selected LLs in the CSA approach needs to trigger (several) expansion WAN path computations which is not necessary in the InA operational mode. Moreover, we observe that the underlying WAN characteristics (i.e., number of nodes and links) impact significantly in the associated path computation functions (i.e., abstraction / expansion) realized by each operational modes. In general, the higher is the WAN size, the more time consuming processes result in both the abstraction computation (done by the InA approach) and the expansion path computation (made by the CSA approach). From a more qualitatively perspective, one can state that the InA approach allows exposing a more accurate abstraction view of the LLs to the 5Gr-SO. This allows the service provider accomplishing a more efficient selection of the virtualized resources when deploying a *NSReq*. On the other hand, the capability of the CSA approach to trigger

dedicated expansion path computations for each selected LL may lead to achieve a better use of the WAN resources.

TABLE I: Comparison of InA and CSA operational modes

WAN scenarios	Num. of RL RA execution	RL RA time (ms)	Num. of abst. LLs	Abst. Info time (ms)	LL deployment time (ms)
Core InA	1	290	18	2683	2595
Core CSA	15	22	8	1859	4708
Core InA	1	77	18	2200	1498
Core CSA	15	14	18	1683	4486

VI. CONCLUSION AND FUTURE WORK

This work proposes two 5Gr-RL operational modes (InA and CSA) within the 5Growth stack. The 5Gr-RL takes over of both: i) providing an abstracted NFVI towards the 5Gr-SO to select the NfviPops and LLs for deploying *NSReq*; and ii) performing LL expansion path computation over the WAN. For both operational modes, the abstraction and expansion functions and their RA algorithms are detailed. The validation of both strategies is experimentally carried out over two NFVIs featuring diverse WANs. This enables to benchmark the 5Gr-RL operational modes through a set of figures of merit. Both approaches present their advantages and drawbacks. InA offers a more accurate abstraction view of the NfviPop connectivity to the 5Gr-SO for the NS placement mechanism. To do this, a more complex and time-consuming 5Gr-RL RA algorithm compared to the CSA strategy is executed. On the other hand, in the CSA approach, the abstracted NFVI towards the 5Gr-SO is simplified relying on a pre-defined set of connectivity service types. This requires afterwards specific expansion path computations at the 5Gr-RL to decide the WAN resources for each selected LL. Expansion path computations tend to increase the overall time for deploying the NS but, at the same time, operate with more granular WAN view. This, in turn, may lead to attain a more efficient use of the WAN resources. Finally, we observed how the WAN size notably impacts on the obtained results. In general, the larger the WAN size is, the higher the RA InA abstraction time becomes as well as the higher the RA CSA expansion computation time is.

Future work aims at providing an exhaustive evaluation of both 5Gr-RL operational modes under dynamic management of heterogeneous NSes with different requirements to further quantify the advantages and drawbacks of each option.

REFERENCES

- [1] A. Zafeiropoulos, et. al., "Enabling Vertical Industries Adoption of 5G Technologies: A Cartography of Evolving Solutions", in Proc. of IEEE EuCNC 2018, June 2018, Ljubljana, Slovenia.
- [2] ETSI GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", v1.1.1, 2014
- [3] C. Papagiani, J. Manges-Bafalluy, et. al., "5Growth: AI-driven 5G for Automation in Vertical Industries," in Proc. of EuCNC 2020, June 2020.
- [4] J. Baranda, L. Vettori, R. Martínez and J. Manges-Bafalluy, "A Mobile Transport Platform Interconnecting VNFs over a Multi-Domain Optical/Wireless Network: Design and Implementation," in Proc. of ONDM 2020, May 2020, Castelldefels (Spain).
- [5] M. Fiorani, et. al., "Transport Abstraction Models for an SDN-Controlled Centralized RAN", IEEE Commun. Letters, Vol.19, Issue. 8, August 2015.
- [6] I. Afolabi, et. al., "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions", IEEE Commun. Surveys and Tutorial, vol. 20, no. 3, 2018.
- [7] N. Sambo, et. al., "Mobile Transport and Computing Platform for 5G Verticals: Resource Abstraction and Implementation", in Proc. of IEEE NFV-SDN 2018, Nov. 2018, Verona, Italy.
- [8] S. Fichera, et. al., "Latency-Aware Resource Orchestration in SDN-Based Packet Over Optical Flexi-Grid Transport Networks", IEEE/OSA J. Optical Commun. and Netw., vol. 11, no. 4, April 2019.
- [9] P. Iovanna, et. al., "Main Challenges on WAN due to NFV and SDN: Multi-layer and Multi-domain Network Virtualization and Routing", in Proc. of ONDM 2015, May 2015, Pisa (Italy).
- [10] A. Mayoral, et. al., "Control orchestration protocol: Unified transport API for distributed cloud and network orchestration", IEEE/OSA J. Optical Commun. and Netw., vol. 9, no. 2, Feb. 2017.
- [11] J. Baranda, et al., "Orchestration of end-to-end network services in the 5G-crosshaul multi-domain multi-technology transport network," IEEE Communications Magazine 56(7), July 2018
- [12] J. Manges, et al., Experimental framework and evaluation of the 5G-Crosshaul Control Infrastructure , Elsevier Computer Standards & Interfaces, January 2019.
- [13] J. Baranda, et. al., "NFV Service Federation: enabling Multi-Provider eHealth Emergency Services", in Proc. of IEEE INFOCOM 2020, July 2020.
- [14] J. Y. Yen, "Finding the k shortest loopless paths in a network", Manage. Sci., vol. 17, no. 11, pp. 712-716, 1971.
- [15] J. Baranda, et. al., "Realising the Network Service Federation vision", in IEEE Vehicular Tech. Mag., Vol.15, Issue. 2, June 2020.
- [16] ETSI GS NFV-IFA 005, "Network Function Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification", August 2018.
- [17] T. Saad, et. al., "Common YANG Data Types for Traffic Engineering", IETF RFC 8776, June 2020.