Integrated deployment prototype for virtual network orchestration solution

Maya Kassis, Massinissa Ait Aba, Hind Castel-Taleb, Maxime Elkael, Andrea Araldo, Badii Jouaber

Integrated Deployment Prototype for Virtual Network Orchestration Solution

Maya Kassis
Telecom SudParis, SAMOVAR, IP-Paris
Evry, FRANCE
Maya_kassis@telecom.sudparis.eu

Massinissa Ait Aba
Davidson Consulting
Paris, France
Massinissa.ait-aba@davidson.fr

Hind Castel-Taleb
Telecom SudParis, SAMOVAR, IP-Paris
Evry, FRANCE
Hind.Castel@telecom-sudparis.eu

Maxime Elkael
Telecom SudParis, SAMOVAR, IP-Paris
Evry, FRANCE
Maxime.Elkael@telecom-sudparis.eu

Andrea Araldo
Telecom SudParis, SAMOVAR, IP-Paris
Evry, FRANCE
Andrea.Araldo@telecom-sudparis.eu

Badii Jouaber
Telecom SudParis, SAMOVAR, IP-Paris
Evry, FRANCE
Badii.Jouaber@telecom-sudparis.eu

***Abstract*—Network slicing in the upcoming Telecom generation is a fundamental feature which is deployed to satisfy the various demands in term of data rate and latency. On the other hand, it is seen as a topic that imposes other questions such as the coexistence of physical and virtual functions. In this context, we consider the resource management problem for 5G networks slicing since the solution searches to optimally allocate multiple Virtual Network Requests (VNRs) on a substrate virtualized physical network. In this demo, we present an integrated framework that uses an agile service platform (Kube5G) to deploy one of the VNE proposed solutions with zero-touch configuration. The aim of this integration is to validate the proposed solution and to practically study the performance differences among multiple algorithms that will be conducted later as well. The overview of the process is shown in steps as exposing the resources' availability of the Physical Nodes (PN), which will be the input of the orchestration algorithm. Successively, the last takes the suitable decision to deploy VNRs on a substrate network, based on the VNRs' demands such as CPU and radio resources and PNs' availability. Afterwards, the decision will be sent to the platform to host the virtual nodes on the chosen physical machines. Bearing in mind that the essential objective of this algorithm is to achieve a better resource usage and increase the VNR acceptance ratio on the physical nodes with respect to the constraints that might affect the performance.**

***Index Terms*—4G/5G, Virtual Network Requests, Network Slicing, Solution Deployment, Kube5G, VNE, Virtualization, OPEX, CAPEX**

I. Introduction

The 5G and beyond telecommunication networks are expected to significantly grow in term of users and to support a various services. Within this framework, it is an essential demand to satisfy all the QoS requirements for all applications. Accordingly, the Network Slicing concept came to the light which defined as a network configuration that allows multiple networks (virtualized and independent) to be created on top of a common physical infrastructure.

This work was supported by the Agence Nationale de la Recherche (ANR) through the AIDY-F2N LabCom Project.

Resource management orchestration is a complex problem with several challenges. This is mainly due to the high variety of applications and services and to the complex topology of networks. One of the main challenges in future networks is how to decide for an efficient embedding of multiple VNRs on the physical network, without affecting the performance of already embedded slices. This involves decisions on embedding several VNRs on the same physical resources. This problem is well known and denoted in the literature, in its simplified form, by the Virtual Network Embedding (VNE) problem. In this paper, we managed to automatically integrate the platform with one of the VNE algorithms [1], which we proposed recently. The goal is to validate the algorithm using real telecommunication services, noting that we will further analyze the result and performance in later work.

However, due to the novelty of 5G and the shortage in its emulation tools, we choose to deploy our work using 4G functions as a first step, assuming that what deployed using 4G architecture, can be deployed later using 5G functions. In addition, the achieved prototype is an improved 4G network as it cannot be realized in the traditional non-virtualized network.

II. Description of the Solution

Our goal of infrastructure providers is to embed a set of VNRs onto a shared physical network in order to maximize the profit and meet QoS requirements of requests. In our work, we consider a dynamic system, where VNRs arrive and leave over time, and must be embedded on a physical network respecting routing and resource constraints. The embedding and release of VNRs are handled by an *agent*, run by the infrastructure provider. Each VNR describes the resources and the VNR needs in the form of a non-oriented graph H^s. On arrival of H^s, the agent embeds it, i.e., embeds every virtual node of H^s onto a physical node. More details on the used data are given in the following.

A. Input parameters

First, we represent the physical network by a non-oriented graph $G(V, E)$. V contains the set of physical nodes. Each physical node $v_j \in V$ is weighted by a maximum CPU capacity C_j. Furthermore, some nodes $v_j \in V$ are weighted by a limited amount of the physical radio resource blocks (RRBs) R_j if they are connected to the remote radio head. E contains the set of the physical edges. Each physical edge $(v_{j_1}, v_{j_2}) \in E$ is weighted by the available bandwidth capacity Bw_{j_1,j_2}, where v_{j_1},$v_{j_1} \in V$. A slice is modeled by a graph $H^s(V^s, E^s)$. Each virtual node $v_i \in V^s$ is weighted by a computational power demand C_i^d. Some nodes $v_i \in V^s$ are weighted by a request amount of the physical radio resource blocks (RRBs) R_i^d. Each virtual edge $(v_{i_1}, v_{i_2}) \in E^s$ is weighted by its bandwidth demand Bw_{i_1,i_2}^d. Since we are in a dynamic system, each slice also has a time of arrival t_a^s and a time of departure t_d^s. The slice is not known before its arrival, thus there is no notion of planning in this model.

Each physical node $v_j \in V$, and physical link $(v_{j_1}, v_{j_2}) \in E$ is associated with a cost $\varsigma(v_j)$ and $\varsigma(v_{j_1}, v_{j_2})$ respectively. Depending on the application, costs may reflect a preference in terms of operator agreements or real cost of operation. These costs can be set higher for a given resource in order to discourage their use (for example to avoid using energy-consuming physical nodes).

Note that C_j, R_j and Bw_{j_1,j_2} are dynamic quantities. Indeed, every time a virtual node $v_i \in V^s$ is mapped into $v_j \in V$, it consumes some resource and thus C_j and R_j (if $R_i^d > 0$) are decremented accordingly. Every time a virtual edge $(v_{i_1}, v_{i_2}) \in E^s$ uses a physical edge $(v_{j_1}, v_{j_2}) \in E$, it consumes some resource and thus Bw_{j_1,j_2} is decremented accordingly. The resources consumed by $v_i \in V^s$ and $(v_{i_1}, v_{i_2}) \in E^s$ are restored after removing the slice $H^s(V^s, E^s)$.

The chosen embedding of the graph H^s in the graph G has to respect some constraints. Each virtual node $v_i \in V^s$ of the slice must be mapped to one physical node $v_j \in V$ respecting the resource constraint ($C_i^d \leqslant C_j$). At most only one virtual node $v_i \in V^s$ of the same slice must be assigned to the same physical node $v_j \in V$. The purpose of this constraint is to reduce failures if a duplicate virtual node of a slice is assigned to the same physical node, and this physical node fails. However, this constraint can be relaxed for some or all physical nodes. Furthermore, each node $v_i \in V^s$ with RRBs requests R_i^d, must be affected to a node $v_j \in V$ connected to the remote radio head, respecting the limited amount of available RRBs R_j. Each virtual edge $(v_{i_1}, v_{i_2}) \in E^s$ must be mapped onto a path of the physical graph, $\{v_0, v_1, ..., v_f\}$, where v_{i_1} is mapped to v_0, and v_{i_2} to v_f. The physical edges of this path must have sufficient available bandwidth, i.e., $Bw_{i_1,i_2}^d \leqslant Bw_{v_j,v_{j+1}} \ \forall j \in [0..f-1]$.

We embed slice by slice a set of slices that arrives over time, our final objective is maximizing the overall acceptance ratio, i.e., the fraction of virtual networks successfully embedded over the entire sequence of virtual network requests. The difficulty is that we don't have in advance the information about the arrival of the slices and the resources needed for each one. So we can only take temporally local decisions, during which we cannot calculate an acceptance ratio. We thus need an objective function that can be calculated at any decision instant and that, in the long term, would lead to maximizing the acceptance ratio. Thus, we take as objective function the minimization of the resources used for embedding each slice. The goal is then to embed each arriving slice with the least resource consumption. We showed in [1] that running our procedure with such an objective function effectively maximizes acceptance ratio.

B. Proposed orchestration algorithm

We use the proposed method in [1], by adding radio constraints. The method consists of two steps:

1) The first phase consists of selecting a reduced number of paths between each two nodes $v_{j1} \in V$ and $v_{j2} \in V$. The goal is to reduce the number of paths, and thus the size of the problem, by heuristically selecting a subset of paths while hopefully formulating a new problem that includes a solution close to the optimal one. In this phase, for a given number K, we select K widest paths between each two nodes $v_{j1} \in V$ and $v_{j2} \in V$. The K-Widest paths Problem is a problem of finding K paths between two nodes of the graph maximizing the bandwidth of the minimum-bandwidth edge in the path. We select the best K paths which provide the largest bandwidth between v_{j1} and v_{j2}.
2) In the second phase, we apply a combinatorial solver to determine near-optimal solutions.

III. Description of the Platform

In our lab, the physical topology consists of 7 geographically distributed machines, including 3 machines in Evry-Telecom SudParis School, and 4 machines in Palaiseau-Telecom Paris School.

We built Kubernetes cluster to form one solid infrastructure network that can be used to run virtual services on it and ensure other important features such as redundancy, fault tolerance and availability. Moreover, we installed Kube5G, which is a cloud native framework that is tailored to telco applications [2]. It is an open source platform allows support both CNF and VNF applications and used to deploy on-demand services in a cloud native environment. To encapsulate the lifecycle management operations of VNFs, an operator framework is introduced to facilitate service automation, achieve zero-touch configuration and update/upgrade dynamic services. The main deployed telco functions are as follows, noting that each virtual function is deployed in a separated pods and they all compose one single slice: Radio Access Network (RAN) controlled by a Software Defined RAN (SD-RAN) controller (FlexRAN), Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving and PDN Gateway for Control plane (SPGWC) and Serving and PDN Gateway for User plane (SPGWU) [3]. However, we programmed a blank SIM to connect it to the

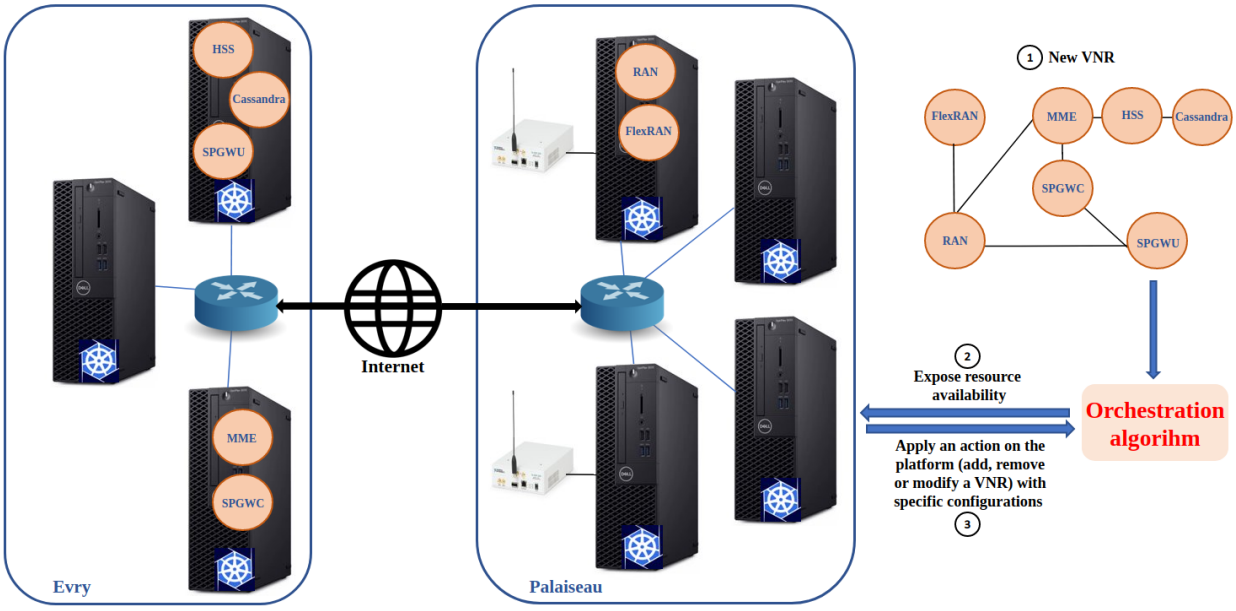

Fig. 1. The stages of VNR deployment in our integrated prototype

emulated telecom slice with the use of USRP card. A video demonstrating the connection has been made available at [4]

Furthermore, we managed to run multiple slices on the physical network, each VN is running on the chosen PN by the orchestration algorithm.

IV. Integration Process

Before starting the cycle of actions, a description of the slice is provided to the algorithm, including the average and the standard deviation of the resource usage of each VNs "CPU and memory" as static values. These values represent the resource demands that should be respected in the physical nodes and symbolizes the space each VN will take from the PNs. Later, we launch our prototype, by doing the following steps regularly each 5 minutes.

1) The platform will provide the algorithm with resource information about the usage of CPU and memory using kubernetes metric server and the available bandwidth between each 2 physical nodes using iperf.
2) Upon an arrive/leave of a slice, the algorithm will calculate the best resources to place the virtual nodes on as well as the CPU and memory limits.
3) The algorithm will inform the platform with the action needed beside the configurations related to it.
4) The platform will form the yaml configuration file and will apply the requested action on the cluster automatically without any human intervention.
5) The Platform will expose the running pods' statistics to observe the actual usage values of each VN.

Figure 1 gives a global view of the used platform, as well as the different deployment stages.

V. Conclusion and Future Perspectives

Our goal in this paper is to demonstrate the automatic and dynamic integration between a VNE solution and kube5G service platform. Moreover, we managed to emulate the algorithm's decisions of placing multiple 4G slices as real and completed services on our physical machines and we are looking forward to analyze the services' performance later. This integration is also feasible with different VNE solutions, as the second step, we are planning to integrate it with the method proposed recently by [5], to analyze its result using the same platform and tool later, which may be more efficient for specific scenarios. Furthermore, considering this work is a part of a whole solution related to 5G network slicing, we intend to use 5G functions as well as 4G functions in our upcoming work. In addition, we are planning to deploy other notions like RAN and bandwidth sharing among multiple services and deploy radio slicing concept using SD-RAN.

References

[1] M. Ait aba, M. Elkael, B. Jouaber, H. Castel-Taleb, A. Araldo and D. Olivier, "A two-stage algorithm for the Virtual Network Embedding problem," 2021 IEEE 46th Conference on Local Computer Networks (LCN), 2021, pp. 395-398, doi: 10.1109/LCN52139.2021.9524969.

[2] O. Arouk and N. Nikaein, "Kube5G: A Cloud-Native 5G Service Platform," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, doi: 10.1109/GLOBECOM42002.2020.9348073.

[3] Frédéric Firmin, "The Evovled Packet Core Architecture" https://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core.

[4] *DEMO of a real virtual telecom operator on k8s cluster through kube5G platform, available at https://youtu.be/Q7CSAiwF64g*

[5] M. Elkael, H. Castel-Taleb, B. Jouaber, A. Araldo and M. Ait Aba, "Improved Monte Carlo Tree Search for Virtual Network Embedding," 2021 IEEE 46th Conference on Local Computer Networks (LCN), 2021, pp. 605-612, doi: 10.1109/LCN52139.2021.9524975.