Optimal Computational Resource Allocation and Network Slicing Deployment in 5G Hybrid C-RAN

Antonio De Domenico*, Ya-Feng Liu†, and Wei Yu‡

*CEA-LETI Grenoble, France
†LSEC, ICMSEC, AMSS, Chinese Academy of Sciences, Beijing 100190, China
‡Department of Electrical and Computer Engineering, University of Toronto, Toronto ON, Canada M5S 3G4
Email: antonio.de-domenico@cea.fr, yafliu@lsec.cc.ac.cn, weiyu@ece.utoronto.ca

***Abstract*—Network virtualization is a key enabler for the 5G systems for supporting the novel use cases related to the vertical markets. In this context, we investigate the joint optimal deployment of Virtual Network Functions (VNFs) and the allocation of computational resources in a hybrid cloud infrastructure by taking into account the requirements of the 5G services and the characteristics of the cloud nodes. To achieve this goal, we analyze the relations between functional placement, computational requirements, and latency constraints, and formulate an integer linear programming problem, which can be solved by using a standard solver. Our results underline the advantages of a hybrid architecture over a standard solution with a central cloud, and show that the proposed mechanism to deploy VNFs leads to high resource utilization efficiency and large gains in terms of the number of slice chains that can be supported by the cloud-enhanced 5G networks.**

I. Introduction

The development of the fifth generation (5G) system is driven by the aim to provide new services characterized by heterogeneous requirements. To achieve this, the research community is defining a flexible architecture, where the network infrastructure is logically split into different instances, i.e., network slices, each designed for a specific service and running in a cloud infrastructure. A network slice is composed of a chain of Virtual Network Functions (VNFs), which represent the software implementation of the traditional network functions (NFs), such as coding/encoding, and can be efficiently reconfigured through the European Telecommunications Standards Institute (ETSI) Management and Orchestration and Network Function Virtualization frameworks [1].

In the current vision for 5G, depending on the network load and service requirements, the available cloud resources can be dynamically allocated across slices. Moreover, the VNF chain in each slice can be split [2], e.g., to improve the resource utilization efficiency or to reduce the end-to-end latency. However, when implementing such a paradigm it is important to consider that the traditional NFs are characterized by tight inter-dependencies [3], which are due to the classical design assumption that all NFs reside in the same fixed location, i.e., at a base station. These inter-dependencies result in very stringent latency constraints on the communication link between the Radio Remote Heads (RRHs) and the cloud.

To deal with these constraints and provide services with low latency requirements, edge clouds can be deployed in the network. Nevertheless, due to the high cost for site acquisition in urban areas, each edge cloud typically has lower computational capacity than a central cloud [4], which reduces the number and types of services that it can support. Therefore, in 5G systems, edge clouds and central clouds will coexist in a hybrid architecture. This calls for orchestration mechanisms that take service requirements and network constraints jointly into account, to enable efficient 5G network slicing.

Recently, [5] investigated the tradeoff between computational and fronthauling costs when optimizing the functional split between the RRH and the central cloud. However, it considered clouds with unlimited capacity. The work [6] focused on a hybrid Centralized Radio Access Network (C-RAN) and investigated the functional split that limits the system power consumption and the bandwidth usage in the link between the edge and the central clouds. It did not consider that each VNF has specific processing and latency requirements. The work [7] considered the VNF deployment problem under computational resource constraints but it did not take the VNF latency requirements into consideration. The work [8] investigated the allocation of VNFs in a hybrid C-RAN. It considered the latency requirements of each VNF; however, it did not take into account that functional splits affect the computational resource requirements. All the above works assumed slices with the same constraints; however, 5G systems need to comply with services with diverse requirements, which determine the computational and latency constraints of each VNF.

In contrast to all previous works, in our analysis, we take into account the type of mobile service associated with each network slice as well as the different requirements of the related VNF chains. Then, we propose a framework for jointly optimizing the computational resource allocation and the deployment of VNF chains with heterogeneous requirements in a hybrid cloud infrastructure.

This work has been partially performed in the framework of the H2020 project 5G-MoNArch co-funded by the EU.

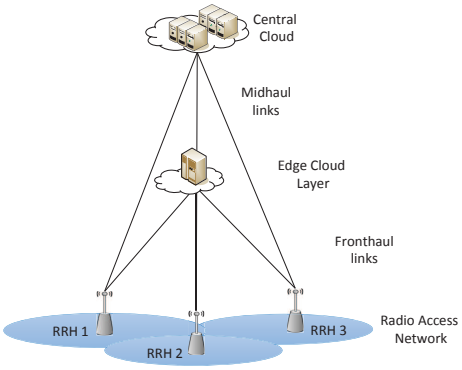

Figure 1. A C-RAN system with a hybrid cloud infrastructure.

II. Network Slices Deployment in a Hybrid C-RAN

We consider a C-RAN system supported by a hybrid cloud infrastructure that enables network slicing (see Fig. 1). A set of RRHs provides access to services with different requirements, such as enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultra-reliable low-latency communications (URLLC). The 5G system maps these services into a set of network slice chains $\mathcal{S} = \{1, 2, \ldots, S\}$, each one composed of physical NFs running in the RRHs and VNFs deployed in the cloud architecture.

The considered hybrid architecture is composed of an edge cloud and one central cloud, with computational capacity [GFLOPS/s] C^e and C^c, respectively. Moreover $d_{e,c}$ represents the distance between the edge cloud and the central cloud and d_e (d_c) indicates the distance of the edge cloud (central cloud) from the RRH where the physical NFs of the served slice chain s is deployed[1]. High-capacity, low-latency fiber links characterized by a speed v ($\sim$200 m/μs) ensure the connectivity between the RRHs and the clouds.

We assume that each service is composed of nine blocks of functions as depicted in Fig. 2: Radio Frequency (RF), lower PHY, higher PHY, lower MAC, higher MAC, lower RLC, higher RLC, PDCP, and RRC [9]. The exact content of these blocks depends on the functional split implementation; in our system, we consider that the RF block is deployed at a RRH, and the other eight functional blocks are virtualized in the cloud infrastructure, thus forming the VNF chain. Each VNF, depending on the associated function and service, has different latency and computational requirements. In particular, we use $\lambda_{s,n}$ to indicate the computational requirement [GFLOPS] for VNF $n \in \mathcal{N}_s := \{1, 2, \ldots, N_s\}$, $s \in \mathcal{S}$, which can be calculated as described in a recent empirical model [10]:

$$\lambda_{s,n} = \frac{C_{\text{exp}} \cdot RB_s}{f_{\text{CPU}}} \sum_{k=0}^{2} \left(\alpha_{n,\text{DL},k} \cdot i^k_{s,\text{DL}} + \alpha_{n,\text{UL},k} \cdot i^k_{s,\text{UL}} \right), \tag{1}$$

where C_{exp} and f_{CPU} are the computational capacity and the frequency of the CPU [GHz] of the machine used for the experiments in [10], RB_s is the number of resource blocks (RBs) allocated to the slice chain s, $i_{s,\text{DL}}$ and $i_{s,\text{UL}}$ are the indices of the modulation and coding schemes (MCSs) of VNF chain s in the downlink (DL) and the uplink (UL) as defined in 3GPP TS 38.214 [11]: the higher the index, the higher the MCS spectral efficiency. Moreover, $\{\alpha_{n,\text{DL},k}, \alpha_{n,\text{UL},k}\}_{n,k}$ are fitting coefficients. The VNFs related to the PHY layer, especially the encoding/decoding, are the most computational demanding functions [2], [10]. Moreover, from (1), the computational requirement of a VNF increases with the spectral efficiency and the number of RBs required by the slice; therefore, services characterized by high data-rate are more computational demanding than low data-rate services.

[1]In this work we do not optimize the association between the VNF chains and the RRHs, and we assume that each chain is connected to a single RRH, which yields well-defined d_e and d_c.

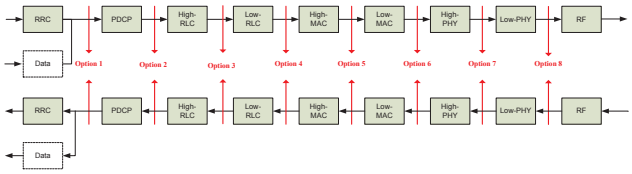

Figure 2. 3GPP options for function split between RRHs and the central cloud [9].

The VNF latency constraints are defined by the interactions in the VNF chain: specifically, VNF n receives inputs from VNF $n-1$, passes its output to VNF $n+1$, and provides a feedback to VNF $n-1$. This process has timing requirements that guarantee reliable operations [3]. Here, we denote with $f_{s,n}$ and $b_{s,n}$ the latency constraints of VNF n with respect to the forward VNF $n+1$ and the backward VNF $n-1$. Among these requirements, the most stringent ones are related to the slot length at the PHY layer [12] and the hybrid automatic repeat request (HARQ) feedback at the MAC layer [3]. In addition, depending on the service, they may be looser or more stringent. For instance, in 5G, the time slot length may be adapted to the service latency requirements.

A virtualized communication system requires the sum of the processing and communication delays to be below these latency constraints. Then, for each $s \in \mathcal{S}$, $n \in \mathcal{N}_s$, we have

$$l^{\text{P}}_{s,n} + f^{\text{C}}_{s,n} \leq f_{s,n} \text{ and } l^{\text{P}}_{s,n} + b^{\text{C}}_{s,n} \leq b_{s,n}, \tag{2}$$

where $l^{\text{P}}_{s,n}$ is the processing latency related to VNF n and $f^{\text{C}}_{s,n}$ and $b^{\text{C}}_{s,n}$ are the latencies related to the communication with the neighbouring VNFs $n+1$ and $n-1$, respectively. The processing latency $l^{\text{P}}_{s,n}$ depends on the computational requirements and the computational rate [GFLOPS/s] allocated to the VNF, and can be modelled as follows:

$$l^{\text{P}}_{s,n} = \frac{\lambda_{s,n} x_{s,n}}{R^c_{s,n}} + \frac{\lambda_{s,n}(1 - x_{s,n})}{R^e_{s,n}}, \; s \in \mathcal{S}, n \in \mathcal{N}_s, \tag{3}$$

where $x_{s,n}$ is a binary (indicator) variable with $x_{s,n} = 1$ if VNF n runs in the central cloud and $x_{s,n} = 0$ if it runs in the edge cloud, $\lambda_{s,n}$ is defined in (1), and $R^c_{s,n}$ and $R^e_{s,n}$ denote the computational rates respectively allocated by the central cloud and the edge cloud to VNF n of chain s.

The first term and the second term in (3) represent the latency experienced by VNF n if it is executed in the central

cloud or in the edge cloud, respectively. Thus, to reduce the processing delay of a VNF, a resource orchestrator may choose to deploy it at the central cloud where more computational resources are available. However, this choice increases the communication latencies ($f^{\mathrm{C}}_{s,n}$ and $b^{\mathrm{C}}_{s,n}$), since the central cloud is typically located in a remote area, far from the access network. This highlights a tradeoff between central and edge clouds and calls for a carefully designed scheme that takes into account the limited computational resources as well as the latency introduced by the link between the central cloud and the edge cloud, and between them and the access network.

In the next section, to optimize the computational resource usage, we investigate the joint optimal computational rate allocation and VNF deployment in a hybrid cloud architecture.

III. Optimal resource allocation and VNF deployment in a hybrid Cloud infrastructure

In this section, we derive the minimum computational rate needed for satisfying each VNF requirement, by fixing the associated (central/edge) cloud node. Then, using this information, we formulate an integer linear programming (ILP) that optimizes the deployment of the VNFs on the clouds.

A. Analysis of the minimum required computational rate

The minimum computational rate required by a VNF to satisfy its latency and computational requirements depends on whether or not it is in the same cloud as its neighbouring VNFs. In general, deploying the entire chain in the same cloud reduces the slice resource footprint, thus increasing the number of services that can be supported by the cloud architecture. Specifically, for any $s \in \mathcal{S}$, the minimum computational rate needed by VNF $n \in \mathcal{N}_s \setminus \{1, N_s\}$, if it is co-located with $n+1$ and $n-1$ is as follows[2]:

$$C_{s,n} = \frac{\lambda_{s,n}}{\min\{f_{s,n}, b_{s,n}\}}.$$

However, if VNFs n and $n+1$ are not in the same node, the computational rate allocated to VNF n may need to be increased to compensate for the *forward* communication delay $\frac{d_{e,c}}{v}$ introduced by the functional split; accordingly, we have

$$C^{+}_{s,n} = \max\left\{\frac{\lambda_{s,n}}{f_{s,n} - \frac{d_{e,c}}{v}}, C_{s,n}\right\},$$

with the constraint that $d_{e,c} < v \cdot f_{s,n}$, i.e., it is not possible to split VNFs n and $n+1$ if the distance between the central cloud and the edge cloud is larger than $v \cdot f_{s,n}$. Similarly, if VNFs n and $n-1$ are not in the same cloud, due to the *backward* communication delay $\frac{d_{e,c}}{v}$, the minimum computational rate is

$$C^{-}_{s,n} = \max\left\{\frac{\lambda_{s,n}}{b_{s,n} - \frac{d_{e,c}}{v}}, C_{s,n}\right\},$$

with the constraint that $d_{e,c} < v \cdot b_{s,n}$, i.e., the distance between the two clouds also limits the possible split between VNFs n and $n-1$. Therefore, when optimizing the slice deployment, the forward and backward functional split constraints must be considered, and they can be explicitly written as follows[3]:

$$\begin{aligned} d_{e,c}\,|x_{s,n} - x_{s,n+1}| &\le v \cdot f_{s,n}, \\ d_{e,c}\,|x_{s,n} - x_{s,n-1}| &\le v \cdot b_{s,n}, \end{aligned} \tag{4}$$

where $|x_{s,n} - x_{s,n+1}|$ is equal to one if there is a functional split between VNF n and VNF $n+1$, and zero otherwise. Likewise, $|x_{s,n} - x_{s,n-1}|$ is equal to one if there is a functional split between VNF n and VNF $n-1$, and zero otherwise. Now, let us define for each $s \in \mathcal{S}, n \in \mathcal{N}_s \setminus \{1, N_s\}$:

$$\begin{aligned} &\Delta C^{-}_{s,n} = C^{-}_{s,n} - C_{s,n};\ \Delta x^{c-}_{s,n} = \max\{x_{s,n} - x_{s,n-1}, 0\};\\ &\Delta C^{+}_{s,n} = C^{+}_{s,n} - C_{s,n};\ \Delta x^{c+}_{s,n} = \max\{x_{s,n} - x_{s,n+1}, 0\};\\ &\Delta x^{e-}_{s,n} = \max\{-x_{s,n} + x_{s,n-1}, 0\};\\ &\Delta x^{e+}_{s,n} = \max\{-x_{s,n} + x_{s,n+1}, 0\}, \end{aligned}$$

where $\Delta C^{-}_{s,n}$ and $\Delta C^{+}_{s,n}$ describe the additional rate, required by VNF n when $n-1$ or $n+1$ are in a different cloud, respectively; $\Delta x^{c-}_{s,n}$ and $\Delta x^{c+}_{s,n}$ indicate respectively if there is a split between VNF n located in the central cloud and its neighbouring VNFs $n-1$ and $n+1$; $\Delta x^{e-}_{s,n}$ and $\Delta x^{e+}_{s,n}$ indicate if there is a split between VNF n located in the edge cloud and its neighbouring VNFs $n-1$ and $n+1$, respectively.

Using these notations, the computational rate to be allocated by the central cloud or by the edge cloud to VNF $n \in \mathcal{N}_s \setminus \{1, N_s\}$ can be computed respectively as follows:

$$\begin{aligned} R^{c}_{s,n} &= C_{s,n} x_{s,n} + \max\left\{\Delta C^{-}_{s,n}\Delta x^{c-}_{s,n}, \Delta C^{+}_{s,n}\Delta x^{c+}_{s,n}\right\};\\ R^{e}_{s,n} &= C_{s,n}(1 - x_{s,n}) + \max\left\{\Delta C^{-}_{s,n}\Delta x^{e-}_{s,n}, \Delta C^{+}_{s,n}\Delta x^{e+}_{s,n}\right\}. \end{aligned}$$

In the above expressions, the first term corresponds to the minimum computational rate to be allocated to VNF n when it is co-located with VNFs $n+1$ and $n-1$; moreover, the second term denotes the additional rate required in case of functional split. Note that, when both VNFs $n+1$ and $n-1$ are processed in a cloud different from the one where VNF n runs, the additional amount of needed resources depends on the most stringent constraint between $f_{s,n}$ and $b_{s,n}$.

The derivations of the minimum computational rate required by the first and last VNFs are special cases of the previous analysis. In particular, considering that for VNF N_s there may exist a split only with VNF $N_s - 1$, the computational rate needed by N_s when $N_s - 1$ is in the same node is as follows:

$$C_{s,N_s} = \frac{\lambda_{s,N_s}}{\min\{f_{s,N_s}, b_{s,N_s}\}}.$$

If $d_{e,c} < v \cdot b_{s,N_s}$, VNFs N_s and $N_s - 1$ may not run in the same cloud node; in this case, the computational rate to process N_s can be computed as

$$C^{-}_{s,N_s} = \max\left\{\frac{\lambda_{s,N_s}}{b_{s,N_s} - \frac{d_{e,c}}{v}}, C_{s,N_s}\right\}.$$

[2]We assume that the communication latency between two VNFs located in the same cloud is negligible.

[3]Note that we have reformulated the above strict inequalities as non-strict ones to guarantee that the corresponding sets are closed. When an equality holds, an infinite computational rate is required to satisfy the VNF latency constraint; however, this will not be selected as a feasible solution due to the limited available computational resources (see (8b) and (8c) further ahead).

Let us denote as $\Delta C^{-}_{s,N_s} = C^{-}_{s,N_s} - C_{s,N_s}$ the additional rate required by VNF N_s when VNF $N_s - 1$ is in a different cloud; then, the computational rate that VNF N_s needs at the central cloud or at the edge cloud is as follows:

$$\begin{aligned} R^{\mathrm{c}}_{s,N_s} &= C_{s,N_s} x_{s,N_s} + \Delta C^{-}_{s,N_s} \Delta x^{c-}_{s,N_s}, \\ R^{\mathrm{e}}_{s,N_s} &= C_{s,N_s}(1 - x_{s,N_s}) + \Delta C^{-}_{s,N_s} \Delta x^{e-}_{s,N_s}, \end{aligned} \tag{5}$$

where $\Delta x^{c-}_{s,N_s} = \max\{x_{s,N_s} - x_{s,N_s-1}, 0\}$ and $\Delta x^{e-}_{s,N_s} = \max\{-x_{s,N_s} + x_{s,N_s-1}, 0\}$ indicate if there is a split between VNF N_s, respectively located in the central or in the edge cloud, and VNF $N_s - 1$.

For VNF 1, the minimum computational rate depends on whether it runs in the edge or central cloud, even if it is co-located with VNF 2. When the two VNFs are in the central cloud, the rate required for VNF 1 is as follows:

$$C^{c}_{s,1} = \max\left\{ \frac{\lambda_{s,1}}{f_{s,1}}, \frac{\lambda_{s,1}}{b_{s,1} - \frac{d_c}{v}} \right\},$$

which highlights that VNF 1 can run in the central cloud only if $d_c < v \cdot b_{s,1}$. Moreover, when VNFs 1 and 2 are in the edge cloud, the computational rate needed for VNF 1 is:

$$C^{e}_{s,1} = \max\left\{ \frac{\lambda_{s,1}}{f_{s,1}}, \frac{\lambda_{s,1}}{b_{s,1} - \frac{d_e}{v}} \right\},$$

where VNF 1 can be deployed in the edge cloud only if $d_e < v \cdot b_{s,1}$. These constraints on the deployment of VNF 1 can be explicitly written as follows:

$$d_c x_{s,1} + d_e(1 - x_{s,1}) \leq v \cdot b_{s,1}. \tag{6}$$

In addition, when $d_{e,c} < v \cdot f_{s,1}$ there may be a split between VNFs 1 and 2; then, the rate required for VNF 1 when it runs in the central cloud or in the edge cloud is respectively:

$$\begin{aligned} C^{c+}_{s,1} &= \max\left\{ \frac{\lambda_{s,1}}{f_{s,1} - \frac{d_{e,c}}{v}}, C^{c}_{s,1} \right\}, \\ C^{e+}_{s,1} &= \max\left\{ \frac{\lambda_{s,1}}{f_{s,1} - \frac{d_{e,c}}{v}}, C^{e}_{s,1} \right\}. \end{aligned}$$

Now, we use $\Delta C^{c+}_{s,1} = C^{c+}_{s,1} - C^{c}_{s,1}$ and $\Delta C^{e+}_{s,1} = C^{e+}_{s,1} - C^{e}_{s,1}$ to indicate the additional rate to be allocated to VNF 1 when VNF 2 is in a different cloud; accordingly, for VNF 1, the minimum computational rate needed at the central cloud or at the edge cloud is as follows:

$$\begin{aligned} R^{\mathrm{c}}_{s,1} &= C^{c}_{s,1} x_{s,1} + \Delta C^{c+}_{s,1} \Delta x^{c+}_{s,1}; \\ R^{\mathrm{e}}_{s,1} &= C^{e}_{s,1}(1 - x_{s,1}) + \Delta C^{e+}_{s,1} \Delta x^{e+}_{s,1}, \end{aligned} \tag{7}$$

where $\Delta x^{c+}_{s,1} = \max\{x_{s,1} - x_{s,2}, 0\}$ indicates if there is a split between VNF 1 located in the central cloud and VNF 2 and $\Delta x^{e+}_{s,1} = \max\{-x_{s,1} + x_{s,2}, 0\}$ denotes if there is a split between VNF 1 located in the edge cloud and VNF 2.

B. ILP formulation

Finally, by leveraging on the analysis developed in the previous section, we formulate the problem of minimizing the computational rate needed to run S VNF chains by optimizing the VNF chain deployment as follows:

$$\min_{\{x_{s,n}\}} \quad \sum_{s\in\mathcal{S}} \sum_{n\in\mathcal{N}_s} R^{c}_{s,n} + R^{e}_{s,n} \tag{8a}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} \sum_{n\in\mathcal{N}_s} R^{c}_{s,n} \leq C^{c}, \tag{8b}$$

$$\sum_{s\in\mathcal{S}} \sum_{n\in\mathcal{N}_s} R^{e}_{s,n} \leq C^{e}, \tag{8c}$$

$$d_{e,c}\,|x_{s,n} - x_{s,n+1}| \leq v \cdot f_{s,n}, \quad s \in \mathcal{S}, n \in \mathcal{N}_s \setminus \{N_s\}, \tag{8d}$$

$$d_{e,c}\,|x_{s,n} - x_{s,n-1}| \leq v \cdot b_{s,n}, \quad s \in \mathcal{S}, n \in \mathcal{N}_s \setminus \{1\}, \tag{8e}$$

$$d_c x_{s,1} + d_e(1 - x_{s,1}) \leq v \cdot b_{s,1}, \; s \in \mathcal{S}, \tag{8f}$$

$$x_{s,n} \in \{0, 1\}, \qquad s \in \mathcal{S}, n \in \mathcal{N}_s, \tag{8g}$$

where (8b) and (8c) denote the computational capacity constraints at the central cloud and at the edge cloud, respectively. Moreover, the functional split constraints (8d)-(8f) limit the VNF deployment such that the allocated computational resources $R^{c}_{s,n}$ and $R^{e}_{s,n}$ satisfy the VNF latency requirements.

Problem (8) is an ILP that can be efficiently solved using a solver such as Gurobi [13]. In particular, constraints involving the absolute value operator can be formulated as linear constraints [14]. The worst-case complexity of globally solving the ILP is dominated by the number of variables as finding an optimal solution may require exhaustive enumeration. Since we consider an architecture with only two clouds, the overall complexity depends on the number of the VNF chains to be deployed. In future works, we plan to design a polynomial-time heuristic based on the special structure of our problem, which can be efficiently used to find a sub-optimal solution when the complexity of solving (8) globally is too large.

IV. Simulation Results

To assess the proposed VNF deployment framework, we consider a network composed of seven macro cells where RRHs provide radio access to multiple slices: eMBB, mMTC, and two types of URLLC. The mMTC services have loose latency constraints and low throughput requirements; the eMBB ones have large throughput and intermediate latency constraints. The first type of URLCC models factory automation services and it has tight latency constraints but relaxed bandwidth demand; in contrast, the second type of URLLC characterizes services such as virtual reality, and it has large bandwidth demand and low latency requirements. The MCS indices and the number of RBs per slice used to compute the VNF computational demand are given in Table I. The values of the other parameters needed for the computational resource model in (1) can be found in [10].

The considered latency requirements $\{b_{s,n}\}$ for each type of slice s and VNF n are shown in Table II. In this work, without loss of generality, we set $f_{s,n} = b_{s,n+1}$. Moreover, unless otherwise stated, we consider that the set of slice chains request $\mathcal{S}$ is composed of an equal number of eMBB

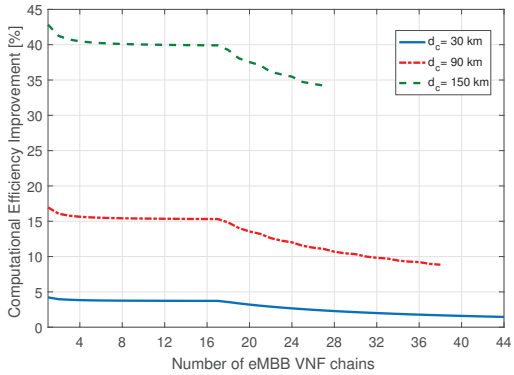

Figure 3. Computational efficiency improvement of the hybrid cloud infrastructure with respect to a C-RAN solution as a function of the number of accepted eMBB VNF chains.

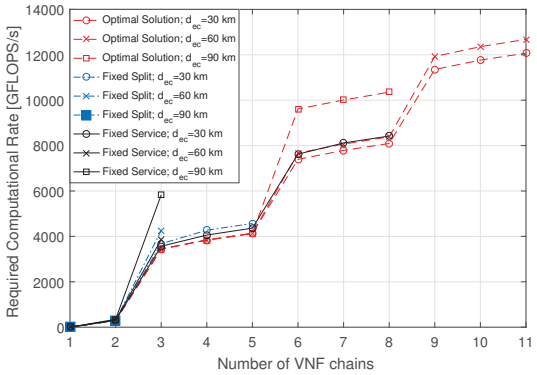

Figure 4. Required computational rate for different VNF deployment schemes with respect to the number of supported VNF chains. C^c=8960 GFLOPS/s and C^e=4480 GFLOPS/s.

and URLLC services; in addition, this set includes only one mMTC (due to its loose service requirements), which is associated with the RRH in the central macro cell. However, the other slice chains are randomly associated with the network RRHs. Moreover, we consider that an Intel Xeon Platinum 8180M Processor is used at the central cloud and an Intel Xeon Silver 4114T Processor is deployed at the edge cloud [15], which is co-located with the central RRH. To conclude, we solve (8) using Gurobi [13], which implements a branch-and-cut algorithm for ILP problems.

Table I
RBs AND DL AND UL MCS INDICES FOR DIFFERENT TYPES OF SERVICES.

	eMBB	mMTC	URLLC 1	URLLC 2
RB_s	250	5	25	500
$i_{s,\mathrm{DL}}$	27	13	27	27
$i_{s,\mathrm{UL}}$	16	8	16	16

Table II
LATENCY CONSTRAINTS FOR VNF AND SERVICE TYPE [3].

	n=1	n=2:4	n=5	n=6	n=7	n=8
$b_{\mathrm{eMBB},n}$ [ms]	1	3	200	500	10^4	$2 \cdot 10^3$
$b_{\mathrm{mMTC},n}$ [ms]	10	10	200	500	10^4	$2 \cdot 10^3$
$b_{\mathrm{URLLC\ 1},n}$ [ms]	0.2	0.2	0.2	0.2	0.2	0.2
$b_{\mathrm{URLLC\ 2},n}$ [ms]	0.5	0.5	0.5	0.5	0.5	0.5

Fig. 3 shows the computational efficiency improvement provided by the hybrid cloud infrastructure with respect to a C-RAN architecture, composed of a single central cloud, as a function of the number of the deployed VNF chains and for different distances of the central cloud from the central macro cell. This metric measures the reduction of computational rate required to support a set of VNF chains led by the hybrid infrastructure with respect to the C-RAN. In this simulation, we focus on the optimal deployment of chains related to a single service (eMBB), to clearly evaluate the advantages of the hybrid architecture over the standard C-RAN approach.

We set the cloud computational capacity C^c equal to 13440 GFLOPS/s in the classic C-RAN architecture; in the hybrid solution, the central cloud has two thirds of the overall capacity, i.e., 8960 GFLOPS/s, while the rest is available at the edge cloud, i.e., C^e=4480 GFLOPS/s. First, we note that, as expected, the larger the distance of the central cloud from the access network, the larger the gain of the hybrid infrastructure. When the distance is equal to 30 km, having an edge cloud leads to limited gains; in fact, the hybrid infrastructure requires only 5% less computational rate as compared to the C-RAN with a central cloud. However, up to 17% and 43% of reduction in terms of required computational rate are achieved when the central cloud is located at 90 km and 150 km from the access network. These improvements come from the relation between communication delay and computational rate requirement in a cloud infrastructure. Deploying a VNF at a distant cloud increases the associated communication latency, which requires an increase in the allocated computational rate in order to satisfy the VNF latency constraints.

For a given distance, the experienced gain slowly varies when the number of chains is low; then, beyond a given number of chains (16 in our results), the edge cloud starts to saturate and the central cloud is used also in the hybrid infrastructure, which notably decreases the measured gains. Finally, it is worth noticing that the improvement in terms of allocated computational rate leads to a larger number of chains that can be supported for a fixed amount of available resources. In fact, although the C-RAN solution deploys up to 38 and 27 eMBB chains when the central cloud is located at 90 km and 150 km; in the same condition, the hybrid infrastructure enables to serve up to 42 and 39 chains.

Now, we consider the scenario with a mix of VNF chains: mMTC, eMMB, and two URLLC services. We compare the performance of the optimal deployment scheme with two baseline solutions denoted as *fixed split* and *fixed service*. In the first solution, the VNFs of each chain, independently of the type of service, are split in the same manner. Specifically, the VNFs up to the lower MAC (see Fig. 2), which have stringent latency and computational requirements, are deployed in the edge cloud, while the other VNFs are instantiated in the central cloud. In contrast, in the fixed service scheme, the mMMTC, eMBB, and URLLC 1 chains are always deployed at the central cloud, while only the URLLC 2 chains (which has the stringest latency constraints) are allotted to the edge cloud.

In Fig. 4 we show the computational rate required by the

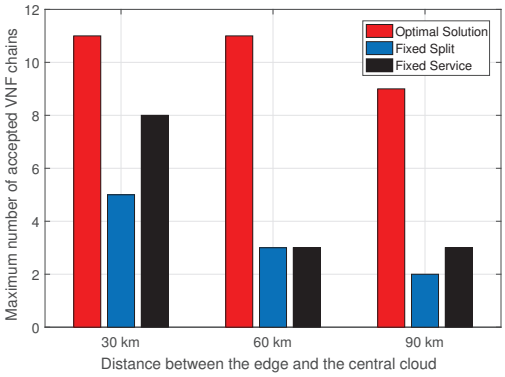

Figure 5. Maximum number of accepted VNF chains as a function of the distance between the edge cloud and the central cloud.

hybrid infrastructure, when using different VNF deployment schemes, as a function of the number of the accepted VNF chains. Dashed, solid, and dotted-dashed lines respectively represent the optimal solution, the fixed service scheme, and the fixed split approach. Moreover, circle marked, cross marked, and square marked lines describe the performance when $d_{e,c}$ is equal to 30, 60, and 90 km, respectively. First, we can notice as the improvement in computational rate of the optimal solution as compared to the baseline schemes increases with the distance between the two clouds: up to 5% and 10% for $d_{e,c} = 30$ km and up to 11% and 19% for $d_{e,c} = 60$ km. For $d_{e,c} = 90$ km, we measure up to 41% gain with respect to the fixed service scheme; however, we cannot measure appreciable gains with respect to the fixed split scheme, since it fails to deploy more than two chains due to the large distance between the two clouds. When the number of VNF chains to deploy is very low or the central cloud is located near the access network, the static schemes have similar performance as that of the optimal solution; however, in the other scenarios, either they require a much larger computational rate or they fail to find a resource distribution that satisfies the service requirements. In contrast, the optimal scheme adapts the functional split at each accepted VNF chain, and when a new request arrives it redistributes the available resources such that the system performance is optimized.

In fact, we can observe from Fig. 5 that the proposed optimal scheme greatly enhances the number of chains that can be successfully deployed with respect to the two static solutions, even when the central cloud is located near the edge cloud (and the macro cell network). Specifically, for $d_{e,c} = 30$ km, the optimal solution provides up 11 VNF chains, while the fixed service and the fixed split achieve up to 8 and 5 chains, with a gain of 37.5% and 120% in terms of the number of deployed chains. These gains further increase when the distance between the central cloud and the edge cloud increases: when $d_{e,c} = 60$ km, the optimal solution can still manage up to 11 VNF chains while the static solutions cannot accept more than 3 VNF chains, which corresponds to a 266% gain in tems of the number of deployed chains. Moreover, when $d_{e,c} = 90$ km, the optimal scheme, the fixed service scheme, and the fixed split scheme accept up to 8, 3, and 2 VNF chains, which correspond to a large gain for the optimal solution as compared to the baseline schemes. Overall, we can observe that, when the number of VNF chains (or equivalently $d_{e,c}$) increases, the proposed scheme brings the desired flexibility to balance the cloud load (i.e., moving VNFs from one cloud to another) and make computational resources available for the chains with more stringent requirements. In contrast, the static schemes lack of such flexibility and lead to limited performance; however, since they are rule-based schemes, they are more scalable than the optimal solution, whose complexity is exponential in the number of the VNFs.

V. Conclusion

In this paper we have investigated the problem of the optimal resource allocation and network slice deployment in a hybrid cloud infrastructure. This problem is analyzed and formulated as an ILP that can be optimally solved through standard solvers. Our results highlight the benefit of a hybrid cloud with respect to a classical C-RAN architecture, composed only of a central cloud, in particular for services with tight latency requirements. Future works will focus on the optimal VNF deployment in an infrastructure composed of multiple edge clouds, and on the development of a heuristic scheme based on the special structure of the problem.

References

[1] ETSI, "GS NFV-IFA 014 – Network functions virtualisation (NFV); Management and orchestration; Network service templates specification," *V2.1.1*, Oct. 2016.

[2] P. Rost *et al.*, "Cloud technologies for flexible 5G radio access networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.

[3] A. Maeder *et al.*, "Towards a flexible functional split for cloud-RAN networks," in *Proc. EuCNC*, June 2014, pp. 1–5.

[4] Nokia White Paper, "The edge cloud: An agile foundation to support advanced new services," *https://onestore.nokia.com/asset/202184*, 2018.

[5] J. Liu *et al.*, "Graph-based framework for flexible baseband function splitting and placement in C-RAN," in *Proc. IEEE ICC*, June 2015, pp. 1958–1963.

[6] A. Alabbasi, X. Wang, and C. Cavdar, "Optimal processing allocation to minimize energy and bandwidth consumption in hybrid CRAN," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 545–555, June 2018.

[7] N. Zhang *et al.*, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017.

[8] O. Arouk *et al.*, "Cost optimization of Cloud-RAN planning and provisioning for 5G networks," in *Proc. IEEE ICC*, May 2018, pp. 1–6.

[9] 3GPP TSG RAN, "TR38.801, Study on new radio access technology: Radio access architecture and interfaces," *V14.0.0*, Mar. 2017.

[10] S. Khatibi, K. Shah, and M. Roshdi, "Modelling of computational resources for 5G RAN," in *Proc. EuCNC*, June 2018, pp. 1–5.

[11] 3GPP TSG RAN, "TS 38.214, NR; Physical layer procedures for data ," *V15.3.0*, Sept. 2018.

[12] ——, "TS 38.202, NR; Services provided by the physical layer," *V15.3.0*, Sept. 2018.

[13] Gurobi Optimization LLC, "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] Intel, "Intel Xeon scalable processors," *https://ark.intel.com/products/series/125191/Intel-Xeon-Scalable-Processors*, Accessed on 10/10/2018.