FluidRAN: Optimized vRAN/MEC Orchestration

Andres Garcia-Saavedra, Xavier Costa-Perez
NEC Laboratories Europe
Heidelberg, Germany

Douglas J. Leith, George Iosifidis
School of Computer Science and Statistics
Trinity College Dublin, Ireland

***Abstract*—Virtualized Radio Access Network (vRAN) architectures constitute a promising solution for the densification needs of 5G networks, as they decouple Base Stations (BUs) functions from Radio Units (RUs) allowing the processing power to be pooled at cost-efficient Central Units (CUs). vRAN facilitates the flexible function relocation (split selection), and therefore enables splits with less stringent network requirements compared to state-of-the-art fully Centralized (C-RAN) systems. In this paper, we study the important and challenging vRAN design problem. We propose a novel modeling approach and a rigorous analytical framework, FluidRAN, that minimizes RAN costs by jointly selecting the splits and the RUs-CUs routing paths. We also consider the increasingly relevant scenario where the RAN needs to support multi-access edge computing (MEC) services, that naturally favor distributed RAN (D-RAN) architectures. Our framework provides a joint vRAN/MEC solution that minimizes operational costs while satisfying the MEC needs. We follow a data-driven evaluation method, using topologies of 3 operational networks. Our results reveal that (*i*) pure C-RAN is rarely a feasible upgrade solution for existing infrastructure, (*ii*) FluidRAN achieves significant cost savings compared to D-RAN systems, and (*iii*) MEC can increase substantially the operator's cost as it pushes vRAN function placement back to RUs.**

I. Introduction

The increasing mobile data traffic and the new demanding services ranging from augmented reality to Industry 4.0 applications, challenge the performance of Radio Access Networks (RAN) and induce unprecedented expenditures to mobile operators. [1] It is apparent today that methods such as the densification of base stations (BS) and the over-provisioning of network links, albeit necessary, cannot address this problem in its entirety; and therefore new RAN solutions are required.

Distributed architectures (D-RAN) used in 3G/4G are cost-inefficient for dense networks due to their expensive radio units, and because they do not facilitate resource pooling. On the other hand, *centralized* RAN (C-RAN) architectures that have recently gained momentum, relocate most BS functions from low-cost Radio Units (RUs) to a central unit (CU) [1]. RUs perform physical-layer tasks and exchange I/Q radio samples with the CUs through the *fronthaul* network. This reduces costs [2]–[5] and improves performance through the central control of tasks such as interference management [6]. However, C-RAN's stringent latency and bandwidth requirements are hard to meet in most RAN deployments nowadays [4], while clean-slate fronthaul designs are very costly [7].

When such pure distributed (D-RAN) or fully centralized (C-RAN) solutions fall short, a **hybrid** RAN design where only some BS functions are centralized might be more suitable [4], [8], [9], [11]. Indeed, we see today a flurry of activities in this space by standardization bodies such as the IEEE 1914 WG and 3GPP RAN3 [12], [13]. These efforts build upon the recent *softwarization* and *cloudification* of C-RAN through SDN/NFV (Software-Defined Networking/Network Function Virtualization) tools. This enables operators to determine the centralization level (*functional split*) of the so-called virtual RAN (vRAN) functions for each RU and in a way that accounts for the available network resources and user demand. This fine-grained network management approach is considered very important for the success of 5G systems [12]–[14]. Nevertheless, *designing the vRAN architecture is a novel and particularly challenging problem*: each configuration has different bandwidth and latency requirements for data transfers across the function locations; involves different amounts of resources (computing power, link capacities); and induces different costs and performance benefits.

Despite the interest of industry and academia [7], [9], [12]–[14] however, we currently lack a methodology for designing vRANs. At the core of this intricate problem lie the decisions for selecting the function splits *and* the CUs-RUs routing paths which, clearly, should be jointly devised. Indeed, the optimal function placement for each BS depends on the capacity and latency of the RU-CU network path, which can only be known after the paths for the entire RAN are selected. On the other hand, finding the optimal path requires knowledge of the flow requirements in terms of volume and latency, which depend on the functional split of each BS. This coupling makes a traditionally challenging routing problem even harder due to the multiple split choices per BS.

The vRAN design problem is further compounded by the advent of (multi-access) edge computing (MEC) [15], a business model where operators lease computing and network resources to vertical sectors, e.g., e-health industry. MEC services target ultra-low latency and high-bandwidth applications and therefore are mainly deployed close to users; and this, in turn, presumes a full-stack D-RAN implementation [16]. Thus, there is an inherent tension between MEC and vRAN which aims at the highest possible centralization of the RAN functions. Given the importance of MEC services (a new revenue source for operators) it is imperative to jointly design them with vRAN, in order to ease this tension and ensure that their performance will meet the expectations set in 5G.

Contributions. In this work we propose **FluidRAN**, a rigorous analytical framework for the optimized design of vRAN networks. We model the BS operation as a *chain of functions* that successively process the traffic to/from the users.

[1] For example, China Mobile reported for 2014 115.1% increase in mobile traffic, but 10.2% profit reduction (Proc. 1st IEEE 5G Summit'15).

Some of these functions (e.g., PDCP in LTE systems) can be implemented in virtual machines (VMs) at the RUs or CUs; while others (e.g., turbo(de)coding in LTE systems) require specific hardware. The function implementation induces a computing cost that may vary across RUs and CUs, and similarly the selected paths affect the data transfer expenses. Our framework yields the vRAN configuration (splits and paths) that minimizes the aggregate operator expenditures.

In order to obtain practical insights, we present and analyze datasets from real backhaul/RAN instances in different countries. We find that these networks do not have a regular structure (e.g., a ring or star topology), exhibit large variation in the RUs-CUs distances, and are highly diverse in terms of link capacities. We then apply our FluidRAN design to these networks and compare the vRAN cost with the respective C-RAN and D-RAN benchmark values. We use measurement-based system parameters (e.g., for the CU/RUs computation costs) and further perform a thorough parameter-sensitivity analysis to characterize their impact on vRAN.

Our contributions can be thus summarized as follows:

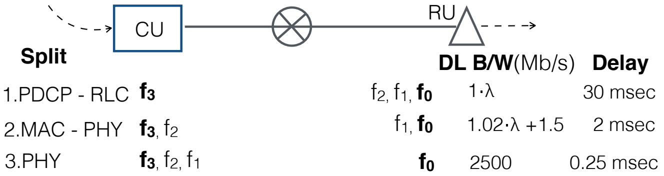
- *Optimization Framework*. To the best of our knowledge, this is the first work introducing an analytical framework for the vRAN design by considering the network and computing resources, and the splits' requirements. Our solution optimizes the placement of vRAN functions jointly with the data routing; and we leverage the Benders' decomposition method to enable its derivation for large systems.
- *Joint vRAN and MEC Design*. We analyze and model the inherent tension among vRAN and MEC. Our framework is extended to jointly decide the placement of MEC services and vRAN functions, yielding a configuration that balances performance benefits and associated costs.
- *Performance Evaluation Using Real Networks*. We analyze 3 backhaul/RAN topologies of different operators, and use market data for costs and 3GPP specs. We show that there is not a one-size-fits-all vRAN configuration and that in practice packetized CPRI-based C-RAN [8] is rarely a feasible solution; on the other hand, FluidRAN, provides significant cost benefits compared to D-RAN.

Paper organization. §II discusses the industry background and our datasets, and introduces the model. §III provides the problem's formulation and §IV its solution method. §V proposes the joint optimization of vRAN and MEC, and §VI presents a thorough data-driven evaluation of FluidRAN. We review the literature in §VII and conclude in §VIII.

II. Model and Problem Statement

A. Background and Data Analysis

There are several levels of centralization [12], but the key splits are those in Fig. 1. Split 1 does not have traffic overheads, enables the co-location of L3 (NAS, RRC, IP) and L2 (PDCP, RLC, MAC) tasks of BSs, and enhances user mobility management. Whenever a CU is available, essentially there is no reason not to have split 1. Split 2 improves hardware utilization, enables multi-cell coordination for CoMP and eICIC, but has significant traffic overheads and an order of magnitude tighter delay bound (for data transfers among function locations). Finally, split 3 (C-RAN) consumes very high bandwidth (which is load-independent), has extremely low delay bounds, but maximizes spectrum efficiency and hardware usage [8]. Finally, regarding the fronthaul, the expensive point-to-point links are expected to be replaced with packet-based shared links [8], [12], [13].



Split			DL B/W(Mb/s)	Delay
1.PDCP - RLC	$\mathbf{f_3}$	$f_2, f_1, \mathbf{f_0}$	$1 \cdot \lambda$	30 msec
2.MAC - PHY	$\mathbf{f_3}, f_2$	$f_1, \mathbf{f_0}$	$1.02 \cdot \lambda + 1.5$	2 msec
3.PHY	$\mathbf{f_3}, f_2, f_1$	$\mathbf{f_0}$	2500	0.25 msec

Fig. 1: Bandwidth and latency requirements of main splits; function f_2 requires f_1, and placement of f_3, f_0 is fixed [12]; λ is the traffic.

We studied the backhaul/RANs from operators in Romania (denoted R1), Switzerland (R2), and Italy (R3), shown in Fig. 2(a)-(c), and we obtained the following insights. First, the *RAN configurations can be very heterogeneous*. The RANs have up to 200 RUs; R3 has only fiber links, R2 mainly wireless links and R1 fiber, copper and wireless links. The networks differ in the number of paths connecting each RU with the CU location. R1 has high path redundancy with a mean (median) value of 6.63 (7) paths, while R3 has often only 1 path (mean 1.6). Clearly, there cannot be a one-size-fits-all RAN split. Second, *these RANs do not have a typical (e.g., tree) structure*. Some RUs are placed as far as 20Km (R3) and 10Km (R2 and R1) while others are in 0.1Km distance from the CU. This renders heuristic or greedy routing policies inefficient.

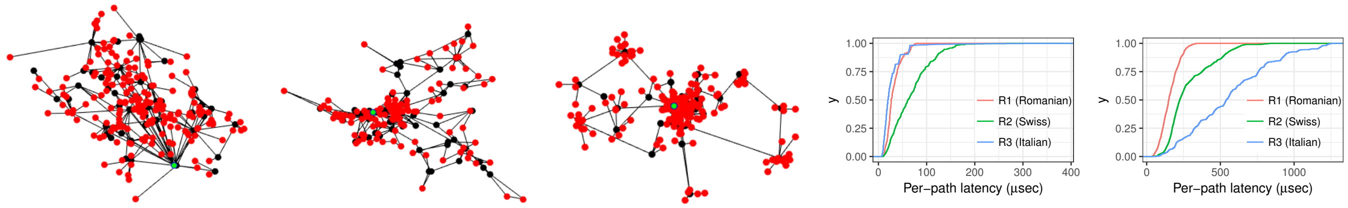
Third, the RUs and CUs are *connected with diverse links* having capacity that ranges from 2000Gb/s down to 2 Gb/s for R3 and 1.25Gb/s for the wireless links of R2. The capacity differences in conjunction with the different link lengths create variation in link and path delays. Fig. 2(d)-(e) presents the eCDF of delays, calculated with a typical store-and-forward switching model that also includes transmission and propagation delays.[2] We observe that: (i) the delay might be up to 40 times higher in some links; (ii) a large number of RUs (different for each RAN) do not support C-RAN (split 3).

B. Model Preliminaries

Fronthaul Network. We consider a RAN with a set $\mathcal{N}$ of N RUs and 1 CU.[3] These are connected through a packet-based network $G = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I}$ is the superset of routers, CU (node 0) and the RUs; and $\mathcal{E}$ is the set of links connecting these elements. Each link $(i,j) \in \mathcal{E}$ has capacity c_{ij} (Mb/s), and introduces delay d_{ij} (secs). Let $p := \{(0,i_1),(i_1,i_2),\ldots,(i_L,n) : (i,j) \in \mathcal{E}\}$ denote a CU-RU n path; $\mathcal{P}_n$ is the set of all RU n paths and $\mathcal{P} = \cup_{n=1}^{N} \mathcal{P}_n$ the set of all CU-RUs paths. Each $p \in \mathcal{P}$ is described by the aggregate delay d_p of its constituent links. There is an average data transfer cost due to consumed energy, leasing

[2] We conservatively used $12000/c_{ij}$, 4μs/Km (cable) or 3μs/Km (wireless), and $5\mu secs$ for transmission, propagation, and processing delay, respectively.

[3] We consider 1 CU, as in practice the RUs are assigned to CUs before split decisions; and most often there is 1 CU location [18].



(a) Romania topology (R1). (b) Swiss topology (R2). (c) Italy topology (R3). (d) Links Delay Distribution (e) Paths Delay Distribution

Fig. 2: (a)-(c): Three actual RANs in Europe: red dots indicate the RUs' locations; black dots the routers/switches; and green dot the CU location which has been placed at the EPC (most central position). (d)-(e) The eCDF of link and path delay in these topologies.

costs, equipment utilization, etc. We denote γ_p the cost for path p (monetary units/byte) and define the *routing cost vector*:

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_{|\mathcal{P}|}) .$$

RAN Functions. The RAN operation is modeled as a chain of 4 functions: f_0, f_1, f_2, and f_3 [12]. Function f_0 corresponds to the basic radio tasks (analog processing, etc.) and it is placed at RUs. Assuming LTE, f_3 corresponds to PDCP and above functions and is always placed at the CU (whenever there is one available). On the other hand, f_2 (RLC and MAC) and f_1 (all PHY functions not in f_0) are placed either at RUs or CU, and this decision is devised independently for each RU. The function placement sets the delay-bandwidth requirements between the CU and each RU, Fig. 1. In vRAN f_2 and f_1 can be implemented in virtual machines (VMs). The cost for initiating and using a VM depends on the hardware. CUs are in central facilities and use high-end servers; hence this cost will be lower compared to RUs [3]. We denote with α_n (monetary units) the average (offset) cost for instantiating a VM in RU n (due to cooling, leasing fees, etc.), with β_n (monetary units per cycle) the average cost for serving each request; and define the respective parameters α_0, β_0 for CU [17]. The *computing cost vectors* are then:

$$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_N), \quad \boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_N) .$$

We denote with ρ_1 and ρ_2 the (nominal) processing loads (cycles per Mb/s) of f_1 and f_2, respectively. Moreover, each RU n and CU have processing capacity P_n and P_0 (cycles), shared by the VMs. When the load is below these bounds, a constant (and very small) processing delay is induced.

Demand. We focus on the downlink but our study can be extended to include uplink. Each RU n serves the requests of users that are generated by an i.i.d. process $\{\Lambda_n(t)\}_{t=1}^{\infty}$, with $E[\Lambda_n(t)] = \lambda_n$ (Mb/s) and we denote the vector $\boldsymbol{\lambda} = (\lambda_n : n \in \mathcal{N})$. The requests at RU n create an aggregate flow emanating from the CU routed to RU n. Hence, the RAN operation can be modeled as a multi-commodity flow problem where the flows depend on the placement of RAN functions.

Fig. 3 depicts the detailed system model.

C. Problem Statement

The objective of the operator is to select the vRAN configuration that will satisfy the users' demand while minimizing the aggregate expenditures. The latter, in such virtualized systems, are mainly due to computing and data routing costs [3], [5]. Several trade-offs arise here. On the one hand, placing the functions at RUs reduces the network's load and hence the routing costs. On the other hand, aggregating the RAN functions at the CU reduces computing costs (economics of scale) and offers centralized control that can improve the network's performance, e.g., through sophisticated interference management techniques. However, some splits have very tight delay constraints and create high fronthaul traffic, while the CU might not have enough computation power to accommodate all RAN functions. The operators' decisions need to be fine-grained, i.e., per RU, and consider all the above aspects. We formally state the RAN design problem as follows:

FluidRAN Design Problem (FRD): Given the anticipated demand $\boldsymbol{\lambda}$; a network $G = (\mathcal{I}, \mathcal{E})$ with links capacities and delays; computing and routing costs, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, determine:

- *service chaining*: where to place functions f_1, f_2 for each RU $n \in \mathcal{N}$ and the CU;
- *service provisioning*: how to route user traffic from the CU to the associated RU;

so as to serve the users request with the minimum cost.

We formalize and solve FRD in §III and IV, respectively and we extend it for the case of MEC services in §V.

III. FluidRAN Design

Function placement. We define the decision $x_n = \{0, 1\}$ to deploy or not, function f_1 in RU n, and the deployment decision $y_n = \{0, 1\}$ for f_2; $x_0, y_0 = \{0, 1\}$ are the respective decisions for the CU. When a function is deployed at the CU it can serve many RUs (if needed). Due to their requirements for specific hardware and their bandwidth-delay needs, f_0 and f_3 are always placed at the RUs and CUs, respectively. We define the *service placement vectors*:

$$\boldsymbol{x} = \big(x_0, x_n \in \{0, 1\} : n \in \mathcal{N}\big), \ \boldsymbol{y} = \big(y_0, y_n \in \{0, 1\} : n \in \mathcal{N}\big)$$

Due to the chain structure of the RAN functions, f_1 cannot be deployed at the CU unless f_2 is also placed there, hence

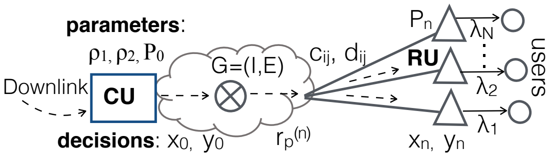


Fig. 3: Detailed system model for FluidRAN.

$x_0 \leq y_0$. Similarly, f_2 cannot be deployed at RU n unless f_1 is there, hence $y_n \leq x_n$. Moreover, we need to deploy each function either at the CU or at the RU, hence:

$$x_n + x_0 \geq 1, \ \text{ and } \ y_n + y_0 \geq 1, \ \ \forall n \in \mathcal{N}$$

where we have inequality constraints (instead of equality) as each RU might implement f_1, f_2, independently of other RUs.

Routing decisions. We denote with $r_p^{(n)}$ the traffic (Mb/s) emanating from CU and routed over path $p \in \mathcal{P}_n$ to RU n, and we define the routing matrix $\boldsymbol{r}{=}(r_p^{(n)}, p{\in}\mathcal{P}, n{\in}\mathcal{N})$. The routing decisions need to respect the link capacities:

$$\sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}_n} r_p^{(n)} I_p^{ij} \leq c_{ij}, \ \forall\,(i,j) \in \mathcal{E},$$

where parameter $I_p^{ij} {\in} \{0,1\}$ indicates whether path p includes link (i, j). Finally, we consider a packet-based network [8], [12] where multiple paths can be selected, as long as it holds:

$$\sum_{p\in\mathcal{P}_n} r_p^{(n)} = S_n, \ \ \forall\, n \in \mathcal{N}\,,$$

where S_n is the data flow for RU n (Mb/s) and depends on the traffic λ_n and the function deployment (see Fig. 1):

$$S_n(x_n, y_n) = x_n Q_{1n} - y_n Q_{2n} + (1 - x_n) Q_3\,, \tag{1}$$

with $Q_{1n} = 1.02\lambda_n{+}1.5$, $Q_{2n} = 0.2\lambda_n{+}1.5$, and $Q_3 = 2500$. Note that when both f_2 and f_1 are deployed at CU ($x_n, y_n = 0$; split 3), the bandwidth is independent of λ_n.

Delay constraints. For each path p the flow $r_p^{(n)}$ cannot be non-zero if its delay d_p exceeds the respective delay threshold [12]. Therefore, the function placement decisions $\boldsymbol{x}, \boldsymbol{y}$ determine which paths are eligible for each split. To capture this dependency, let us first partition the set of paths as follows: set $\mathcal{P}_n^A \subseteq \mathcal{P}_n$ of paths with delay larger than 30msec; set $\mathcal{P}_n^B \subseteq \mathcal{P}_n$ of paths with delay larger than 2msec; and set $\mathcal{P}_n^C \subseteq \mathcal{P}_n$ of paths with delay larger than 0.25msec. Obviously, it holds $P_n^A \subseteq P_n^B \subseteq P_n^C$. Then, for split 1 ($x_n = y_n = 1$) we need to set all flows in paths of $\mathcal{P}_n^A$ equal to zero, for split 2 ($x_n = 1, y_n = 0$) set equal to zero all flows in $\mathcal{P}_n^B$, and for split 3 ($x_n = y_n = 0$) zeroize all flows in $\mathcal{P}_n^C$.

Objective function. The goal of the network operator is to minimize its costs while satisfying the users' demand. For the data transfer average cost we consider a basic linear function:

$$U_p(r_p^{(n)}) = \gamma_p \sum_{n\in\mathcal{N}} r_p^{(n)}, \ \ p \in \mathcal{P}\,. \tag{2}$$

The computing costs depend on the functions placement. When f_1 and f_2 are deployed at RU n the cost is:

$$V_n(\boldsymbol{x}, \boldsymbol{y}) = \alpha_n(x_n + y_n) + (\beta_n \rho_1 \lambda_n) x_n + (\beta_n \rho_2 \lambda_n) y_n\,, \tag{3}$$

and when they are deployed at the CU:

$$\begin{aligned} V_0(\boldsymbol{x}, \boldsymbol{y}) = \alpha_0(x_0 + y_0) + \beta_0\rho_1 \sum_{n\in\mathcal{N}} \lambda_n(1 - x_n) + \\ + \beta_0\rho_2 \sum_{n\in\mathcal{N}} \lambda_n(1 - y_n)\,, \end{aligned} \tag{4}$$

where the last terms indicate that when an RU does not implement a function this load is shifted to CU. Note that for the objectives we consider constant (yet, different) cost parameters as the loads (routing or processing) are strictly confined by the respective capacity bounds. This standard approach [16], [17], [24] is also validated by measurements (see §VI). We thus can formulate the FDR problem:

Problem 1 (FluidRAN Design Problem: FRD)**.**

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{r}} J_F = V_0(\boldsymbol{x}, \boldsymbol{y}) + \sum_{n\in\mathcal{N}} V_n(\boldsymbol{x}, \boldsymbol{y}) + \sum_{p\in\mathcal{P}} U_p(r_p^{(n)}) \tag{5}$$

$$\text{s.t.} \quad x_0 \leq y_0\,, \ \ y_n \leq x_n, \qquad \forall n \in \mathcal{N} \tag{6}$$

$$x_n + x_0 \geq 1, \ \ y_n + y_0 \geq 1, \qquad \forall n \in \mathcal{N} \tag{7}$$

$$\lambda_n(x_n\rho_1 + y_n\rho_2) \leq P_n, \qquad \forall n \in \mathcal{N} \tag{8}$$

$$\sum_{n=1}^{N} \lambda_n(\rho_1(1{-}x_n){+}\rho_2(1{-}y_n)) {\leq} P_0 \tag{9}$$

$$x_n,\, y_n,\, x_0,\, y_0 \,\in \{0, 1\}, \qquad \forall n \in \mathcal{N} \tag{10}$$

$$\sum_{p\in\mathcal{P}_n} r_p^{(n)} = S_n(x_n, y_n), \qquad \forall n \in \mathcal{N} \tag{11}$$

$$\sum_{p\in\mathcal{P}_n^A} r_p^{(n)} \leq M(2 - x_n - y_n), \qquad \forall n \in \mathcal{N} \tag{12}$$

$$\sum_{p\in\mathcal{P}_n^B} r_p^{(n)} \leq M(1 - x_n + y_n), \qquad \forall n \in \mathcal{N} \tag{13}$$

$$\sum_{p\in\mathcal{P}_n^C} r_p^{(n)} \leq M(x_n + y_n), \qquad \forall n \in \mathcal{N} \tag{14}$$

$$\sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}_n} r_p^{(n)} I_p^{ij} \leq c_{ij}, \qquad \forall (i,j) \in \mathcal{E} \tag{15}$$

$$r_p^{(n)} \geq 0 \qquad \forall p{\in}\mathcal{P}_n, \forall n{\in}\mathcal{N}\,. \tag{16}$$

where $M >> 0$ and (12)-(14) capture the split and path-delay coupling explained above. FRD's complexity is discussed next.

Theorem 1. ***FRD Complexity****. FRD is NP-hard to solve.*

Proof. In a multi-dimensional multiple-choice Knapsack problem ($MMKP$) [19], there are n groups of items and m types of resources; each group i has l_i items; each item j of group i has value v_{ij} and requires r_{ijk} units of type-k resource. The goal is to pick one item from each group so as to maximize the value of collected items subject to the constraints for each resource. We show next a polynomial reduction of our FRD $MMKP \leq_P FRD$ to this Knapsack version via restriction. Consider an instance of FRD where (i) all path delays are very small; (ii) all link capacities exceed the traffic; (iii) and $U_p = 0$, $\forall p$. For every x_n, y_n, $n \in \mathcal{N}$, we can trivially find a solution r_p. We now introduce the binary *configuration* variables: $u_n, v_n, w_n, n \in \mathcal{N}$, with $u_n = 1$ if f_1, f_2 are placed at RU n; $v_n = 1$ if f_1 is placed at RU n and f_2 at CU; and $w_n = 1$ if both f_1, f_2 are placed on CU. Obviously, it should hold $u_n + v_n + w_n = 1$ for every n (representing the Knapsack variables), while the processing constraints with upper bounds P_0 and $P_n, \forall n$ (representing the Knapsack constraints) can be written as linear combinations of these variables, and the same holds for the objective function (for fixed routing). Then, FRD becomes a $MMKP$ problem with $l_i = 3, \forall i$, $m = N + 1$, and if we could solve the former in polynomial time we could solve the latter as well. □

Next, we introduce a solution method for FRD. It is important to stress at this point that our model and solution are generic and can be easily extended for scenarios where, e.g., the routing cost function is strictly convex on the data volume [23], the processing cost is piecewise linear on λ, and so on.

IV. Solution Methodology: Asking Benders' Help

For large networks, FRD computational complexity increases substantially. It is thus important to devise a methodology that expedites its solution. To this end, we leverage the Benders' decomposition method [20] that separates FRD in smaller subproblems: one with the "complicated" variables and one with the continuous variables. This decomposition yields two meaningful subproblems, namely the **routing optimization** problem and the **function placement** problem.

A. Benders' Method at-a-glance

We briefly overview Benders' method [21] using an abstraction of the FRD problem (with a slight abuse of notation):

$$\text{(P)}: \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y},\boldsymbol{r}\in\mathcal{R}} \quad c_1^T\boldsymbol{x} + c_2^T\boldsymbol{y} + c_3^T\boldsymbol{r}$$
$$\text{s.t.} \quad A\boldsymbol{x} + B\boldsymbol{y} + \Gamma\boldsymbol{r} \leq K,$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of the integer variables with $\mathcal{X}$ and $\mathcal{Y}$ the respective feasibility sets, i.e., (6)-(10); $\boldsymbol{r}$ the continuous variables constrained in $\mathcal{R}$ defined by (15)-(16); and A, B, Γ and K the matrices of constraints coupling the discrete and continuous variables, i.e., (11)-(14). The main idea in Benders' method is to use the equivalent formulation:

$$\min_{\bar{\boldsymbol{x}}\in\mathcal{X},\bar{\boldsymbol{y}}\in\mathcal{Y}} \big\{c_1^T\bar{\boldsymbol{x}} + c_2^T\bar{\boldsymbol{y}} + \min_{\boldsymbol{r}\in\mathcal{R}}\{c_3^T\boldsymbol{r} : \Gamma\boldsymbol{r} \leq K - A\bar{\boldsymbol{x}} - B\bar{\boldsymbol{y}}\}\big\}.$$

The inner minimization problem, which is defined for fixed variables $\boldsymbol{x} = \bar{\boldsymbol{x}}$ and $\boldsymbol{y} = \bar{\boldsymbol{y}}$, is an LP solvable in polynomial time. By the *strong duality* theorem [27], we use its dual:

$$\min_{\bar{\boldsymbol{x}}\in\mathcal{X},\bar{\boldsymbol{y}}\in\mathcal{Y}} \big\{c_1^T\bar{\boldsymbol{x}} + c_2^T\bar{\boldsymbol{y}} + \qquad (17)$$
$$+ \max_{\boldsymbol{\pi}\in\boldsymbol{R}_+^m}\{\boldsymbol{\pi}^T(K - A\bar{\boldsymbol{x}} - B\bar{\boldsymbol{y}}) : \boldsymbol{\pi}^T\Gamma \leq c_3\}\big\}.$$

This formulation reveals the method's philosophy. We substitute the inner maximization problem (*slave*) with a continuous variable θ. In each iteration τ, the outer (*master*) problem is solved and yields the lower bound $LB^{(\tau)}$ for (P) and also $\boldsymbol{x}^{(\tau)}$ and $\boldsymbol{y}^{(\tau)}$ which are used in the slave problem. The latter gives an upper bound $UB^{(\tau)}$ for (P) and a set of *cuts*, i.e., constraints for θ, $\boldsymbol{x}$ and $\boldsymbol{y}$. These are added in the master problem which this way is refined and gives an improved bound $LB^{(\tau+1)}$. These iterations terminate when the upper and lower bounds become equal, i.e., the optimal solution is reached. The method's gist is the replacement of the large set of variables $r_p^{(n)}$ and constraints in (P) with a single variable θ and dual constraints that are gradually added, and the solution is often obtained before the full constraint set is reconstructed.

B. Decomposition of FRD

Applying this idea to FRD, we iteratively place the functions and then optimize routing for this configuration. In each round the J_F is improved. In detail, the *slave* routing problem is obtained if we fix the binary variables of FRD to $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$:

$$P_S(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}): \min_{\boldsymbol{r}\geq\boldsymbol{0}} \sum_{p\in\mathcal{P}}\gamma_p \sum_{n\in\mathcal{N}} r_p^{(n)} \qquad (18)$$
$$\text{s.t.} \quad \sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}_n} r_p^{(n)} I_p^{ij} \leq c_{ij}, \qquad \forall\,(i,j)\in\mathcal{I} \qquad (19)$$
$$\sum_{p\in\mathcal{P}_n} r_p^{(n)} = S_n(\bar{x}_n,\bar{y}_n), \qquad \forall\, n\in\mathcal{N} \qquad (20)$$
$$\sum_{p\in\mathcal{P}_n^A} r_p^{(n)} \leq M(2-\bar{x}_n-\bar{y}_n), \qquad \forall n\in\mathcal{N} \qquad (21)$$
$$\sum_{p\in\mathcal{P}_n^B} r_p^{(n)} \leq M(1-\bar{x}_n+\bar{y}_n), \qquad \forall n\in\mathcal{N} \qquad (22)$$
$$\sum_{p\in\mathcal{P}_n^C} r_p^{(n)} \leq M(\bar{x}_n+\bar{y}_n), \qquad \forall n\in\mathcal{N} \qquad (23)$$

The dual of P_S can be succinctly written as follows:

$$P_{SD}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}): \max_{\boldsymbol{\pi}} \quad g(\boldsymbol{\pi},\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) \ \text{ s.t. } \boldsymbol{H}^T\boldsymbol{\pi} \leq \boldsymbol{\gamma}, \qquad (24)$$

where $\boldsymbol{\gamma} = (\gamma_p, p\in\mathcal{P})$, $\boldsymbol{H}$ is set by the objective and constraints (19)-(23), $\boldsymbol{\pi}$ is the matrix of the $|\mathcal{E}| + 4|\mathcal{N}|$ dual variables (one for each constraint in P_S), and the dual function:

$$g(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\boldsymbol{\pi}) = \sum_{(i,j)\in\mathcal{I}} c_{ij}\pi_{1ij} + \sum_{n\in\mathcal{N}} S_n(\bar{x_n},\bar{y_n})\pi_{2n} -$$
$$\sum_{n\in\mathcal{N}} M(2-\bar{x}_n-\bar{y}_n)\pi_{3n} - \sum_{n\in\mathcal{N}} M(1-\bar{x}_n+\bar{y}_n)\pi_{4n} -$$
$$\sum_{n\in\mathcal{N}} M(\bar{x}_n+\bar{y}_n)\pi_{5n} \qquad (25)$$

.

The *master* function placement problem is:

$$P_M(\mathcal{C}_1,\mathcal{C}_2): \min_{\boldsymbol{x},\boldsymbol{y},\theta} V_0(\boldsymbol{x},\boldsymbol{y}) + \theta + \sum_{n\in\mathcal{N}} V_n(\boldsymbol{x},\boldsymbol{y}) \qquad (26)$$
$$\text{s.t.} \quad x_0 \leq y_0\,, \quad y_n \leq x_n, \qquad \forall n\in\mathcal{N} \quad (27)$$
$$x_n + x_0 \geq 1, \ \ y_n + y_0 \geq 1, \qquad \forall n\in\mathcal{N} \quad (28)$$
$$\lambda_n(x_n\rho_1 + y_n\rho_2) \leq P_n, \qquad \forall n\in\mathcal{N} \quad (29)$$
$$\sum_{n=1}^{N}\lambda_n(\rho_1(1-x_n)+\rho_2(1-y_n))\leq P_0 \qquad (30)$$
$$g(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\pi}^m) \leq \theta, \qquad \forall\boldsymbol{\pi}^m\in\mathcal{C}_1 \quad (31)$$
$$g(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\pi}^l) \leq 0 \qquad \forall\boldsymbol{\pi}^l\in\mathcal{C}_2 \quad (32)$$
$$\theta\geq 0, \ \ x_0,y_0\in\{0,1\}, \ \ x_n,y_n\in\{0,1\}, \qquad \forall n\in\mathcal{N} \quad (33)$$

FDA algorithm is summarized in Algorithm 1. It is executed iteratively. In each iteration τ, we first solve the *master* problem in order to obtain the currently optimal configuration decisions $\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}$ and the value of the surrogate variable $\theta^{(\tau)}$ (step 3). This gives the current lower bound $LB^{(\tau)}$ (step 4). Then, we solve the dual of the *slave* problem P_{SD} using as input the current variables $\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}$ (step 5). Accordingly we update the upper bound, which is set by the value of the relaxed master problem (step 6). Finally we add a proper cut in the set of cuts $\mathcal{C}_1$ if the dual optimal value is bounded, or in $\mathcal{C}_2$ if the dual is unbounded (which corresponds to a ray of the dual). These steps are repeated until the upper and lower bounds coincide. If FRD is unfeasible we will obtain an

```
Algorithm 1: (FDA) FRD Decomposition Algorithm
Initialize: $\tau = 1$; $\mathcal{C}_1^{(0)} = \mathcal{C}_2^{(0)} = \emptyset$; $UB^{(0)} = -LB^{(0)} >> 1$.
repeat
    Solve problem $P_M(\mathcal{C}_1^{(\tau)}, \mathcal{C}_2^{(\tau)})$ to obtain $\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}, \theta^{(\tau)}$.
    Set $LB^{(\tau)} = V_0(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}) + \theta^{(\tau)} + \sum_n V_n(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)})$.
    Solve problem $P_{SD}(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)})$ to obtain $\boldsymbol{\pi}^{(\tau)}$.
    If $UB^{(\tau)} < UB^{(\tau-1)}$ then $UB^{(\tau)} =$
        $V_0(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}) + g(\boldsymbol{\pi}^{(\tau)}, \boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}) + \sum_n V_n(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)})$.
    If $g(\boldsymbol{\pi}^{(\tau)}, \boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)}) < \infty$ then
        $\boldsymbol{\pi}^m \leftarrow$ extreme point
        $\mathcal{C}_1^{(\tau+1)} = \mathcal{C}_1^{(\tau)} \cup \{\boldsymbol{\pi}^m\}$.
    If $g(\boldsymbol{\pi}^\tau, \boldsymbol{x}^\tau, \boldsymbol{y}^\tau) \to \infty$ then
        $\boldsymbol{\pi}^l \leftarrow$ extreme direction/ray
        $\mathcal{C}_2^{(\tau+1)} = \mathcal{C}_2^{(\tau)} \cup \{\boldsymbol{\pi}^l\}$.
    $\tau = \tau + 1$.
until $UB^{(\tau)} - LB^{(\tau)} \to 0$;
Set the optimal configuration as $\boldsymbol{x}^* = \boldsymbol{x}^{(\tau)}$ and $\boldsymbol{y}^* = \boldsymbol{y}^{(\tau)}$.
Compute the optimal routing $\boldsymbol{r}^*$ by solving $P_{SD}(\boldsymbol{x}^{(\tau)}, \boldsymbol{y}^{(\tau)})$.
```

unbounded value for the slave problem in the first iteration.

Note that the *master* problem remains intricate (at least as hard as the MMKP), but its dimension has been substantially reduced as we have replaced all the routing variables with θ. There are several methods to solve it. Among the most efficient approaches is to remove the integrality constraints (LP relaxation) for variables $x_0, y_0, x_n, y_n \forall n$ for the iterations of Algorithm 1 until the $UB - LB$ gap is reduced enough. This will make P_M a problem solvable in polynomial time. Then, when we approach to an optimal solution (i.e., $|(UB - LB| \to 0$), we need to re-introduce these integrality constraints so as to obtain an optimal feasible solution. This method is proved to preserve the optimality of the problem, since no optimal solutions are removed by the cuts added during the iterations that use the relaxed version of P_M, and at the same time significantly expedites the execution of Algorithm 1, e.g., see [26]. The next theorem describes the FDA performance.

Theorem 2. ***Optimality of Algorithm FDA****. The algorithm converges to the optimal solution of the FRD problem in a finite number of iterations.*

Proof. The proof follows from the Partition Theorem in [20]. Applying this result in our case, we see that the solution of the FRD problem can be obtained from the equivalent problem (using the abstract notation of problem (17), Section IV-A):

$$\min_{\boldsymbol{x}, \boldsymbol{y}, \theta} \; c_1^T \boldsymbol{x} + c_2^T \boldsymbol{y} + \theta \quad \text{s.t.} \;\; (\boldsymbol{x}, \boldsymbol{y}, \theta) \in \mathcal{G}\,, \tag{34}$$

where $\mathcal{G}$ is the set of constraints for all variables, created by the intersection of the constraints in $\mathcal{X}$, $\mathcal{Y}$ and the convex hull of the extreme halflines stemming from the dual slave problem (which is a polyhedral cone $\mathcal{C}$). The algorithm starts with the minimal set of constraints $\mathcal{G}^{(0)}$ (for $\mathcal{C}_1 = \mathcal{C}_2 = \emptyset$) and at each iteration τ adds one extreme halfline of the cone $\mathcal{C}$ in $\mathcal{G}^{(\tau)}$ by modifying the sets $\mathcal{C}_1^{(\tau)}$ and $\mathcal{C}_2^{(\tau)}$. Given that there are finite such constraints (depending on the dimension of matrix Γ in problem (17)), and since in each iteration we add a different halfline, the algorithm terminates in a finite number of steps. The convergence to the optimal solution is ensured by the fact that, in the worst case, we will reconstruct the initial set $\mathcal{G}$. □

V. FluidRAN and Edge-Computing Services

We explore how FluidRAN is designed when the network accommodates a MEC service [15], which ideally should be placed at RUs. A MEC service impacts the RAN design for the following reasons: (i) might have tighter delay needs (even 1ms [15]) than split 1 (30ms) or split 2 (2ms); (ii) increases the traffic load for splits 1 and 2, but not for D-RAN (as the MEC traffic is served at RUs) and interestingly not for split 3 (load-independent); (iii) increases the computation load and renders centralization more challenging. In §II.A and Fig. 2(d-e) we see that real RANs are both delay and capacity constrained, and hence *MEC can significantly affect the eligible splits.*

Our framework is tailored to facilitate the joint design of C-RAN and MEC services. From a modeling perspective, MEC is a function, say f_4, added on the service chain, i.e., its placement presumes the co-location of f_1, f_2 and f_3. We denote with $z_n \in \{0, 1\}$ the decision to place f_4 (and necessarily f_3 as well[4]) at RU n ($z_n = 1$); and with $z_0 \in \{0, 1\}$ the decision to have a MEC service at the CU (that can serve more than one RUs). Let $\boldsymbol{z} = (z_n, n \in \mathcal{N})$ be the MEC placement vector, and λ_n^M (Mb/s) the MEC demand at RU n. Clearly, it should hold:

$$y_0 \le z_0, \quad z_n \le y_n, \;\; z_0 + z_n \ge 1, \;\; \forall\, n \in \mathcal{N} \tag{35}$$

Moreover, the processing constraints need to account for f_4:

$$\lambda_n(x_n\rho_1 + y_n\rho_2) + \lambda_n^M z_n \rho_4 \le P_n, \;\; \forall\, n \in \mathcal{N}, \tag{36}$$

$$\sum_{n=1}^{N} \lambda_n\rho_1(1 - x_n) + \lambda_n\rho_2(1 - y_n) + \lambda_n^M \rho_4(1 - z_n) \le P_0, \tag{37}$$

where ρ_4 denotes the processing load of both f_4 and f_3. Let us now discuss three different representative MEC scenarios.

Strict Delay Constraints. When the MEC service has tight delay requirements, constraints (12)-(14) must be modified in order to ensure only eligible paths are allowed for each configuration. Also, eq. (36)-(37) replace (8)-(9) in FRD, and the new constraint (35) is added. Finally the value of the flow bound S_n has to be modified accordingly:

$$S_n(x_n, y_n, z_n) = x_n[1.02(\lambda_n + \lambda_n^M) + 1.5] - z_n(\lambda_n^M) - \\ y_n[0.02(\lambda_n + \lambda_n^M) + 1.5] + (1 - x_n)2500.$$

Finally, FRD will have a new objective, say $J_{FM}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{r}, \boldsymbol{z})$, which includes the MEC cost (for f_4) in V_0 and V_n.

Delay Cost Component. When the MEC service exhibits instead some elasticity in delay, the constraints are not affected but J_{FM} needs to include a delay cost component, thus it is $J_{FM} =$

$$V_0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) + \sum_{n \in \mathcal{N}} V_n(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) + \sum_{p \in \mathcal{P}} U_p(r_p^{(n)}) + D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}),$$

[4] Some simple MEC services (e.g., signal estimators) do not interact with the data plane and can be placed before PDCP; then, z_n refers only to f_4.

where $D(\cdot)$ is an application-specific cost that increases with the MEC experienced delay, which in turn depends on the functions placement and the traffic load.

Centralization - Delay Balance. Finally, some operators might wish to balance (i.e., fine-tune) the centralization they achieve and the MEC delay cost that this induces. In this case the objective of FRD can be augmented by the addition of an explicit centralization benefit function, i.e., $\delta \cdot \sum_n (1-x_n) + (1-y_n) + (1-z_n)$ which rewards the design with δ units for each function deployed at the CU. By selecting an appropriate value for this parameter, the operator can balance the MEC benefits (from reducing delay) and C-RAN costs from placing functions to the RUs instead of the CU.

VI. Performance Evaluation

We present extensive experiments for the evaluation of FluidRAN in 10 (sub)figures. Our goal is to:

- Apply FluidRAN to the 3 real-world RANs of Fig. 2;
- Evaluate the impact of computing cost and capacity on FRD, and compare these results with D-RAN and C-RAN;
- Examine how the routing cost and the traffic load affect the centralization level, i.e., the optimal RAN split per BS;
- Study the interplay of vRAN with MEC.

A. Methodology and Experiments Setup

In order to obtain realistic results, we use reference values for the system parameters from prior measurement-based studies which are also complemented by our own lab measurements. Furthermore we have conducted a thorough sensitivity analysis for the parameters, beyond their reference values.

We parametrize our model conservatively, with 1 user/TTI, 20MHz BW (100 PRBs), 2x2 MIMO, CFI=1, 2 TBs of 75376 bits/subframe, and IP MTU 1500 B, that is, assuming a high-load scenario $\lambda = 150$Mb/s for each BS. We consider a single Intel Haswell i7-4770 3.40GHz CPU core as our unit of CPU capacity (*reference core*, RC). From our own measurements and those reported in [22], we estimate that, in relative terms, f_3 is responsible for 20% of the total consumption of a software-based LTE BS, f_2 consumes 15%, and f_1 up to 65%. From [5], we calculate the (absolute) computing needs of a software-based LTE BS. In our scenario a BS would require 750 μs of the reference CPU core to process each 1-ms subframe [5] which means a 75% CPU consumption; hence, we set $\rho_1 = 3.25$ and $\rho_2 = 0.75$ RCs per Gb/s, respectively. Finally, we set $P_0 = 100$ RCs and sufficient computing on each RU to run a full-stack BS, i.e., $P_n = 1$ RC $, \forall n \in \mathcal{N}$.

In practice, estimating computing and routing costs is difficult as they depend on the employed hardware, leasing agreements, and so on. We note however that the function placement and routing decisions are essentially affected by the relative values of the computing cost parameters across RUs and CU (α_0, β_0 and α_n, β_n), as well as the ratios of computing over routing costs (γ). Hence, in the following we estimate and use such relative values for $\boldsymbol{a}$, $\boldsymbol{\beta}$ and γ. According to [3], the equipment cost of a D-RAN BS is estimated to \$50K whereas the respective cost of a C-RAN BS (i.e., RU with Split 3, Fig. 1) is \$25K. Based on this information, we assume that the function instantiation cost is approximately half when done in the CU, i.e., $\alpha_0 = \alpha_n/2$; and we set, unless otherwise stated, $\alpha_n = 1 \,\forall n \in \mathcal{N}$, i.e., homogeneous RUs, to ease the analysis. Regarding the processing costs, the main advantage of the CU compared to RUs comes from the pooling gains (cooling, CPU load balancing, etc.). Based on [6], we estimate the CU processing cost to $\beta_0 = 0.017\beta_n$ (linear regression in [6, Fig.6a]). If we take as reference the processing cost at RU, then $\beta_0 = 0.017$ and $\beta_n = 1$.

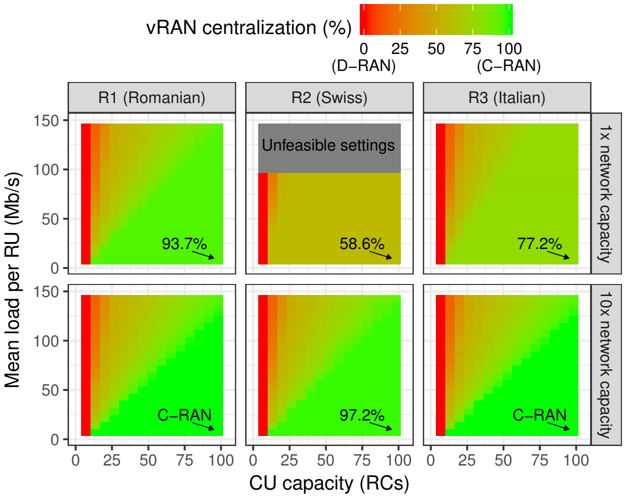


Fig. 4: Ratio of RAN centralized functions in Swiss, Romanian and Italian topologies for different values of CU capacity and traffic load.

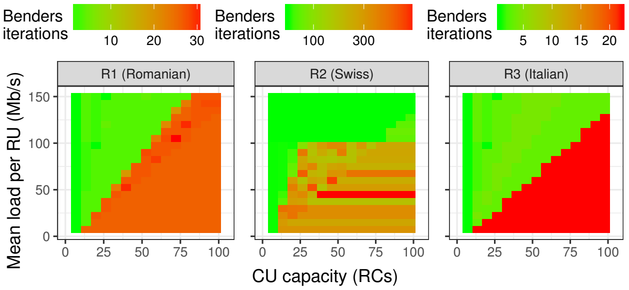


Fig. 5: Number of Benders iterations in Swiss, Romanian and Italian topologies for different values of CU capacity and traffic load.

B. Results

1) Centralization Level and Split Selection: Fig. 4 and Fig. 5 depict the percentage of BS functions f_1 and f_2 placed at the CU (centralized) and the number of Benders iterations till convergence, respectively, in the three topologies of Fig. 2. The results are plotted for an exhaustive set of combinations of CU computing capacity and BS load (λ). We observe that full centralization (C-RAN) is not possible in any of these systems. R2 has the smallest percentage of functions that can be placed at the CU, maximum of 58.6%. This is rather expected as it includes low-capacity wireless links. This under-provisioning is further evinced by the fact that no solution is feasible (not even D-RAN) when the RU load is larger than $\lambda = 100$ Mb/s. On the other hand, R1 achieves 93.7% centralization, even for high traffic (given sufficient CU computing capacity). In the lower plots, we have (artificially) boosted the links' capacity. We see now that both R1 and R3 can achieve full centralization

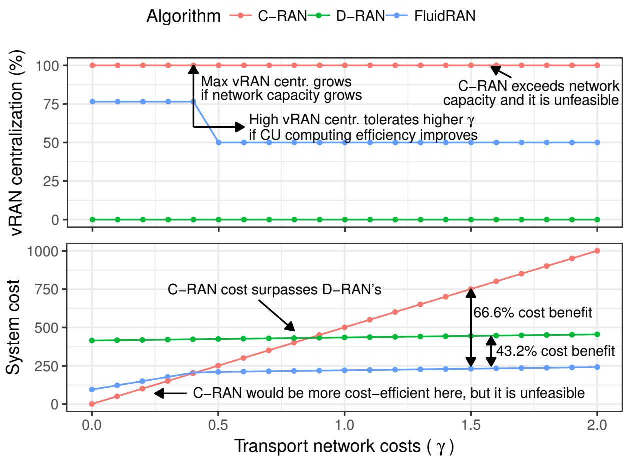


Fig. 6: RAN centralization (top) and system cost (bottom) for Italian topology (R3) for $\alpha_0 = \alpha_n = 1\,\forall n \in \mathcal{N}$ and variable transport costs, for C-RAN, D-RAN and FluidRAN architectures.

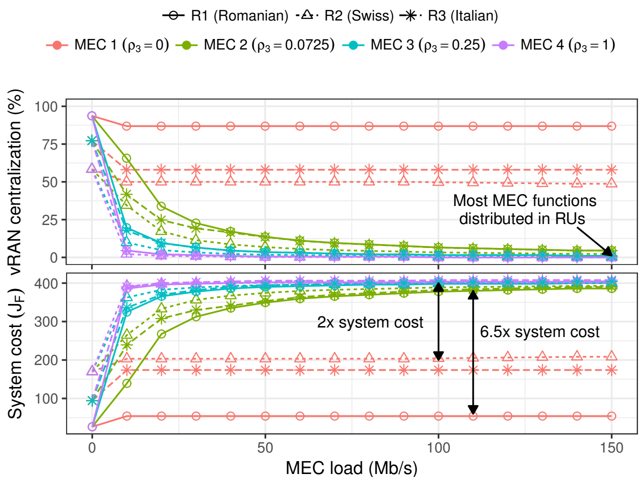


Fig. 7: RAN centralization (top) and cost $J_F(\boldsymbol{x},\boldsymbol{y},\boldsymbol{r})$ (bottom) for different MEC process characteristics and loads. Non-MEC load is $\lambda = 10$ Mb/s for all RUs; $P_0 = 100$ RCs and $P_n = 2$ for each RU.

(for high CU capacity), and R2 also centralizes 97.2% of the functions. This numerical test reveals that centralization in R1-R3 is mainly constrained by the links' capacity.

2) Impact of Parameters on vRAN Cost: We next perform a parameter sensitivity analysis using R3 (Italian topology). We first study the impact of routing cost on vRAN. Fig. 6 shows both the percentage of centralized RAN functions and system costs, when $\alpha_n = \beta_n = 1\,\forall n \in \mathcal{N}$ and $\alpha_0 = \beta_0 = 1$ which is the worst-case scenario where the CU has no computing efficiency advantage compared to RUs. The routing cost ranges from $\gamma = 0$ (no cost) to $\gamma = 2$ (Gb/s)$^{-1}$ (twice the computation cost). Note that γ is defined with reference to computing costs in order to facilitate comparisons. We compare FluidRAN with D-RAN and C-RAN deployments. The latter two are special cases of FRD where the function placement variables are fixed, i.e., routing is still optimized. We stress that the latter is not implementable in these systems (as shown in Fig. 4) but the respective cost is shown for comparison purposes.

Let us focus on the top plot of Fig. 6. For low routing costs, i.e., $\gamma < 0.25$, FluidRAN finds in maximizing the amount of functions that are centralized (in this case 77.2%) the most cost-efficient solution. Clearly, even for $\alpha_n = \alpha_0$ and $\beta_n = \beta_0$, centralization is beneficial due to aggregation (less instantiations costs in CU). If we focus on the bottom plot we observe that, as we increase γ, there is a point where FluidRAN and C-RAN yield the same cost ($\gamma \sim 0.37$). If we further increase γ, the most cost-efficient configuration is to lower the amount of centralization to 50% (split 2 for all RUs). This reduces the amount of traffic in the network compensating in this way the high computational costs of RUs. Noticeable, the system cost of C-RAN overpasses traditional RAN when $\gamma > 1$. Finally, note that improving the computing efficiency at CU (i.e., decrease α_0/α_n) ensures high centralization even for large γ; and improving the links' capacity increases the maximum centralization.

3) Tension between vRAN and MEC: Finally, we analyze the impact of MEC on the cost and centralization of the 3 topologies. To this aim, we consider 4 services that differ on their computation needs: MEC 1 ($\rho_4 = 0$) and MEC 4 ($\rho_4 = 1$) are two extreme cases, MEC 2 ($\rho_4 = 0.0725$) and MEC 3 ($\rho_4 = 0.25$) mimic the computational needs of an optimization application and a virtual reality application experimentally assessed in [17] and [28], respectively. In order to highlight the impact of MEC on the vRAN operation, we plot the cost only for the latter (i.e., J_F instead of J_{FM}), and for the same reason we set $\gamma = 0$.

Fig. 7 depicts the centralization and system cost of FluidRAN for different MEC loads $\lambda n^M = \lambda^M, \forall n$. Observe that as the MEC load λ^M increases, vRAN centralization is reduced in order to alleviate the saturated links. This effect is pronounced for computation-intensive MEC, since these services consume also the available CU computing capacity. Interestingly, computing-intensive MEC can increase multiple times (e.g., 2 times in R2 and 6 times in 6.5 times in R2) the system's expenditures. This increase is not only due to the new processing demand, which is obvious factor and hence not depicted in the figure, but also because vRAN must yield centralization gains when faced with heavy MEC services. Finally, note that for very high MEC loads all networks opt for D-RAN and have similar costs J_F (since they have similar number of RUs and $\gamma = 0$).

VII. Related Work

Cloud-RAN and Beyond. Initially, C-RAN abbreviated the term "Centralized-RAN" describing a system where some BSs' hardware was collocated; later this evolved so BSs functions could run in common hardware (Cloud-RAN) [1]. Many works analyzed the cost-efficiency gains of this [2]–[4], [6], [7]. C-RAN raises unprecedented challenges [4], [7] and this has spurred many efforts to address them. For example, [11] proposed BS functional splits different from pure C-RAN in order to relax the delay and bandwidth constraints for the network. Other works analyzed the cost-benefits trade-offs of the splits [4], [12]–[14]; while [8], [9] (among others) studied the impact of packetization in C-RAN splits. These works show that a more flexible split in packet-based networks is possible. We go here a further step and analyze how this split should be designed in conjunction with routing policies.

Multi-access edge computing. MEC has recently been introduced to enable new use cases from vertical industries such as automotive sectors [15]. However, the relevant literature on the topic has focused on the optimization at the user side, e.g., [16], while our goal is to decide how to deploy MEC services within the RAN, from the operator's perspective.

Network Architectures and Virtualization. The softwarization of cellular networks renders them similar to cloud computing or content delivery systems. Our problem is related (among others) to the joint server selection and data routing problem in ISP-CDN networks [23]; and we know that such joint optimizations outperform respective independent policies [24]. FRD is a more intricate problem due to the functions' chaining. This, in turn is related to virtual network embedding and VNF design problems [25]. Joint routing and split selection was studied in [10] via heuristics. Nevertheless, in FRD the functions placement impacts both the routing (by changing the data transfer needs) and the processing costs. Our novel analysis caters for these intricacies.

Modeling and Solution. FRD is a challenging problem due to the coupling of function placement and routing in the constraints and its objective. For the latter we used an average cost model as is suitable for such network design problems. We employed Benders' decomposition [20] that gives an exact solution, a method attracting increasing interest [21]. To the best of our knowledge, this is the first time it is applied in a problem with function chaining. There are several methods for expediting Benders' algorithm which can be directly applied here [21]. Other possible solutions include facility location methods, which however require stylized models, and Lagrange decompositions [27] which do not guarantee optimality.

VIII. Conclusions

The design of RAN lies at the center of our research efforts to deliver the next generation of 5G systems. To this end, the latest proposals of industry consortia and standardization bodies focus on the cloudification of RAN, flexible functions split, and on packet-based fronthaul routing. Building on these suggestions, we provide a rigorous mathematical modeling for the virtualized RAN (vRAN) design problem and a solution methodology that takes into account its key practical aspects. The main take-away conclusion of our analysis is that a *flexible vRAN design (FluidRAN) which jointly selects the function split and routing policy, tailored to the available network and computing resources, is the optimal way to proceed.*

Indeed, using data from actual RAN instances of three different operators, we showed that upgrading to fully-fledged C-RAN is most often infeasible, while D-RAN induces higher costs than vRAN. FluidRAN achieves the maximum vRAN centralization by selecting the optimal split and routing path *for each RU/CU pair*. This fine-grained design approach is imperative as *our data analysis showed that, in practice, RAN networks can be highly diverse* with high or low link capacities, single or multiple CU-RU paths, and different routing costs. The design of FluidRAN is further perplexed when the RAN has to support MEC services which favor decentralized configurations and increase substantially the system costs. Our framework enables the joint vRAN/MEC design and hence allows operators to select the vRAN solution that meets their needs, by balancing costs and benefits.

Acknowledgments

This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 671598 (5G-Crosshaul project) and 761536 (5G-Transformer project), and from Science Foundation Ireland (SFI) under Grant Number 17/CDA/4760.

References

[1] Y. Lin et al., "Wireless Network Cloud: Architecture and System Requirements", *IBM Journal of Research and Dev.*, vol. 54, 2010.

[2] K. C. Garikipati, K. Fawaz, G. K. Shin, "RT-OPEX: Flexible Scheduling for Cloud-RAN Processing", *in Proc. of ACM CoNEXT*, 2016.

[3] V. Suryaprakash, et al., "Are Heterogeneous Cloud-Based Radio Access Networks Cost Effective?", *in IEEE JSAC*, vol. 33, no. 10, 2015.

[4] A. Checko *et al.*, "Evaluating C-RAN Fronthaul Functional Splits in Terms of Network Level Energy and Cost Savings," *Journal of Communications and Networks*, vol. 18, no. 2, 2016.

[5] N. Nikaein, "Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling", *in Proc. of ACM MCS*, 2015.

[6] P. Rost, et al., "The Complexity-Rate Tradeoff of Centralized Radio Access Networks", *IEEE Trans. on Wireless Comm.*, 14 (11), 2015.

[7] C-Lin I., Y. Yuan, J. Huang, S. Ma, C. Cui, R. Duan, "Rethink Fronthaul for Soft RAN", *IEEE Communications Magazine*, vol. 53, no. 9, 2015.

[8] C. Chang, N. Nikaein, T. Spyropoulos, "Impact of Packetization and Functional Split on C-RAN Fronthaul Performance", *IEEE ICC*, 2016.

[9] C. Chang, et al., "FlexCRAN: A Flexible Functional Split Framework Over Ethernet Fronthaul in Cloud-RAN", *in Proc. of IEEE ICC*, 2017.

[10] A. Garcia-Saavedra, et al., "WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul", *IEEE Transactions on Mobile Computing*, to appear.

[11] U. Dotsch et al., "Quantitative Analysis of Split Base Station Processing and Determination of Advantageous Architectures for LTE", *Bell Labs Tech. Journal*, vol. 18, no. 1, 2013.

[12] Small Cell Forum, "R6.0. Small cell Virtualization Functional Splits and Use Cases", *Document 159.07.02, Release 7*, Jan. 2016.

[13] 3GPP, "Study on New Radio Access Technology: Radio Access Architecture and Interfaces", TR 38.801, 2016

[14] IEEE 1914 WG "IEEE WG, Next Generation Fronthaul Interface"

[15] ETSI, "Multi-Access Edge Computing", Online: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing, 2017.

[16] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing", *IEEE Trans. on Signal and Inf. Proc. over Networks*, vol. 1, no. 2, 2015.

[17] J. Kuo et al, "Service Chain Embedding with Max Flow in SDN & Application to Next-Gen Cellular Networks" *IEEE INFOCOM* 2017.

[18] K. Kiyoshima, et al., "Commercial Development of LTE-Advanced: Applying Advanced C-RAN", *NTT Docomo Techn. Journ.*, 17(2), 2015.

[19] M. M. Akbar, M. S. Rahmanb, M. Kaykobadb, E. G. Manninga, G. C. Shojaa,, "Solving the Multidimensional Multiple-choice Knapsack Problem by Constructing Convex Hulls", *Computers and Operations Research*, vol. 33, 2006.

[20] J. F. Benders, "Partitioning Procedures for Solving Mixed-Variables Programming Problems", *Numer. Math*, vol. 4, 1962.

[21] R. Rahmaniani, et al., "The Benders Decomposition Algorithm: A Literature Review", *European Journal of Oper. Res.*, vol. 259, 2017.

[22] C. Y. Yeoh *et al.*, "Performance study of LTE experimental testbed using OpenAirInterface," in *Proc. of ICACT 2016*, Jan 2016, pp. 617–622.

[23] W. Jiang, et al., "Cooperative Content Distribution and Traffic Engineering in an ISP Network", *ACM SIGMETRICS/Performance*, 2009.

[24] G. Rodolakis, S. Siachalou, and L. Georgiadis, "Replicated Server Placement with QoS Constraints", *IEEE Trans. on PDS*, 17(10), 2006.

[25] R. Cohen, L. Eytan, J. Naor, and D. Raz, "Near Optimal Placement of Virtual Network Functions", *in Proc. of IEEE INFOCOM*, 2015.

[26] L. Qian, et al., "Joint BS Association and Power Control via Benders' Decomposition", *IEEE Trans. on Wireless Comm.*, vol. 12, no. 4, 2013.

[27] D. P. Bertsekas, "Network Optimization: Continuous and Discrete Methods", *Athena Scientific*, ISBN: 1-886529-02-7, 1998.

[28] Maxwell, Douglas, et al. "Make large-scale virtual training a reality." *in Proc. of MODSIM 2014*, 2014.