Optimizing allocation and positioning in a disaggregated radio access network aware of paths through the core infrastructure

Felipe Freitas Fonseca[1], **Sand Luz Correa**[1], **Kleber Vieira Cardoso**[1]

[1]Instituto de Informática – Universidade Federal de Goiás
Goiânia – Goiás – Brasil

fonsecafel@gmail.com, {sand,kleber}@inf.ufg.br

Abstract. *Future wireless communication infrastructures, starting from 5G, will operate their radio access networks (RANs) based on virtualized functions distributed over a crosshaul, i.e., a transport solution integrating fronthaul and backhaul. Optimizing the resource allocation and positioning of the virtual network functions of a virtualized RAN (vRAN) is crucial to improve performance. In this paper, we propose a new optimization model to deal with vRAN functions allocation and positioning that seeks to maximize the level of centralization. Our model explores several representative functional splits, including the fully distributed remote unit (RU), while taking into account the limit imposed by the communication paths between the crosshaul and the core network. We compare our model with a state-of-the-art solution and show how our approach improves the centralization level in the majority of the scenarios, even considering the limit imposed by the core infrastructure. Our model also provides higher number of feasible solutions in most of the cases. Additionally, we investigate the positioning of the central unit (CU) and show that its colocation with the core infrastructure is rarely the best choice.*

1. Introduction

Radio access network (RAN) architectures in 3G/4G systems are commonly distributed, i.e, base stations (BSs) and network links are geographically deployed to satisfy the long-term demand estimated during the network planning phase. In a (pure) distributed RAN (D-RAN) architecture, the baseband unit (BBU) and the remote radio head (RRH) of each BS are colocated. D-RAN also implies that each BS is responsible for processing all its wireless traffic, which involves the communication with each user equipment (UE) under its coverage but also the coordination with other BSs for management purposes, e.g., interference control. Additionally, each BS is responsible for routing the downlink/uplink traffic of its users from/to the core network through the backhaul. This approach tends to be easy to scale, but it is cost-inefficient because resource pooling is rare or nonexistent, leaving most of the BSs underutilized.

Cloud RAN (C-RAN) was proposed [Lin et al. 2010] as an alternative approach in which the BBUs may be virtualized and centralized. By pooling resources and deploying only RRHs in the cell sites, a C-RAN approach presents noticeable improvements on energy efficiency and infrastructure maintenance [Checko et al. 2015]. Many works have analyzed these cost-efficiency gains [Suryaprakash et al. 2015, Checko et al. 2016, Rost et al. 2015]. Other works have investigated different issues related to C-RAN. For

example, in [de Lima and Couto 2018], the authors formulated a mixed integer linear programming problem to choose the placement of the radio functions in a C-RAN composed of different levels of hierarchy. In [Chang et al. 2016, Chang et al. 2017], the authors investigated the impact of packetization on the C-RAN fronthaul links taking into consideration the functional splits.

However, a pure C-RAN depends on an expensive fronthaul with high-speed links between each RRH, also known as radio unit (RU), and the central unit (CU) where the BBUs are kept. Thus, a hybrid approach [de Souza et al. 2018, Garcia-Saavedra et al. 2018a, Garcia-Saavedra et al. 2018b, Asensio et al. 2016] has been investigated as the most promising solution for 5G systems, but it depends on some assumptions. First, the several tasks performed in a RAN can be disaggregated into multiple functional splits [Rost et al. 2014], which can be virtualized and run at different locations, under specific constraints. Second, there is an integrated transport solution [Costa-Perez et al. 2017], named as crosshaul, that can operate simultaneously as fronthaul and backhaul.

The hybrid approach for RAN, also known as virtualized RAN (vRAN), introduces some new complex optimization problems, e.g., 1) which are the most cost-effective sites to be updated in order to support vRAN [de Souza et al. 2018], 2) how to schedule channels when the vRAN is shared by multiple providers [Chen et al. 2018], 3) how to maximize the vRAN centralization level taking into account jointly the functional splits and the routing over the crosshaul [Garcia-Saavedra et al. 2018a, Garcia-Saavedra et al. 2018b]. The latter problem is also the focus of this paper.

In [Garcia-Saavedra et al. 2018b], the authors propose two heuristics solutions: 1) a nearly-optimal backtracking scheme, and 2) a low-complex greedy approach. In addition to using non-optimal strategies, part of the problem is simplified, since all RUs served by the same CU employ the same functional split. In [Garcia-Saavedra et al. 2018a], the authors proposed a modeling approach, FluidRAN, that minimizes RAN costs by jointly selecting the splits and the RUs-CU routing paths. They showed that pure C-RAN is rarely a feasible upgrade solution for existing infrastructure and FluidRAN achieves significant cost savings in comparison with D-RAN. However, FluidRAN disregards the possibility of having all RAN functions running in an RU, which may be very common in real-world networks, mainly during the transition to 5G systems. This simplifies the routing problem, since there is no direct traffic between RUs and the core, only between RUs and CU. As a consequence, the authors evaluated only when the CU is collocated with the core.

We propose a new approach, named as *plasticRAN*, to solve the problem of maximizing the vRAN centralization level taking into account jointly the functional splits and the routing over the crosshaul, which derived the following contributions:

- **New problem formulation**: our model considers additional functions distribution among RU and CU, including the possibility of running all functions in an RU. This functional split, which represents 'no split', has no delay constraint. We also introduce the potential paths between RUs and the core network. Naturally, this makes our problem harder to solve.
- **Improved vRAN performance**: the higher granularity in terms of functional splits and the 'no split' option opens opportunities for improving the centralization level and reducing the number of infeasible solutions. Our solution overcomes

FluidRAN in most of the scenarios.

- **New insight on the problem**: we evaluate different locations for CU and core network, including collocated and non-collocated options. Contrary to what can be inferred in other works, including FluidRAN, the non-collocated option offers the best results in most of the scenarios. This happens because not all the traffic needs to flow through the core since the CU is in another place.

The paper is organized as follows. Section 2 briefly describes the RAN functional splits and their demands in terms of latency and bandwidth. In Section 3, we define the system model, which includes the RAN functions, BS and transport nodes, and routing. Section 4 presents the problem formulation as a Mixed Integer Linear Programming (MILP) problem. In this section, we also introduce our solution, based on Bender's Decomposition. In Section 5, we present the performance evaluation, which includes a comparison with FluidRAN. We present our final thoughts and future work in Section 6.

2. RAN Functional Splits

An important issue in vRAN is the disaggregation of the RAN functionalities into multiple functional parts, so that these parts can be virtualized and allocated to different locations. This disaggregation is usually achieved by defining clear interface points, called *splits*, each one having its own bandwidth and delay requirements on the crosshaul.

One way to split the RAN functionality is at the protocol level [Garcia-Saavedra et al. 2018b]. For example, the LTE (Long-Term Evolution) protocol stack is composed of RRC, PDCP, RLC, MAC, and physical (PHY) layers. A RRC-PDPC split in this protocol stack means that all processing from RRC up is centralized, while all processing from PDCP down runs at the edge. Due to their nature, MAC and PHY layers can be further split in order to provide more flexibility. Figure 1 shows all possible splits for LTE and how they divide the protocol stack.

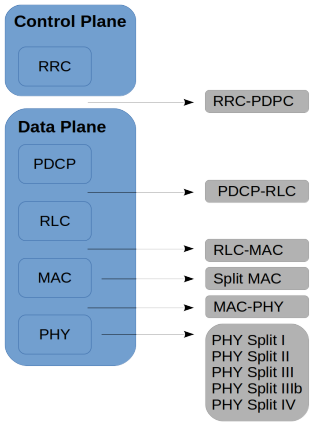

Figure 1. Splits in the LTE stack.

Use Case	One-way latency	DL bandwidth	UL bandwidth
RRC-PDCP	30ms	151Mbps	48Mbps
PDCP-RLC	30ms	151Mbps	48Mbps
RLC-MAC	6ms	151Mbps	48Mbps
Split MAC	6ms	151Mbps	49Mbps
MAC-PHY	250μs	152Mbps	49Mbps
PHY split I	250μs	173Mbps	452Mbps
PHY split II	250μs	933Mbps	903Mbps
PHY split III	250μs	1075Mbps	922Mbps
PHY split IIIb	250μs	1966Mbps	1966Mbps
PHY split IV	250μs	2457.6Mbps	2457.6Mbps

Table 1. Delay and bandwidth requirements for each split, considering 150Mb/s DL and 50Mb/s UL demand [Small Cell Forum 2016].

Table 1 shows delay and bandwidth requirements for each split in LTE. There is a clear trade-off between higher levels of centralization – which yields to gains such as enablement of interference coordination mechanisms, computational resource pooling and on demand scalability of resources – and more stringent network (latency and bandwidth) requirements. For example, PHY split IV is equivalent to pure C-RAN and presents the

toughest network requirements. As the amount of centralization is relaxed, moving toward to RRC-PDCP split, the network requirements are reduced. Therefore, a key issue in the vRAN design is jointly selecting the proper functional splits and the routing paths across the crosshaul. In the next sections, we detail our approach to solve this problem.

3. System Model

RAN functions. The splits in Table 1 fall into three major categories in relation to delay requirements, i.e., those with 30ms, 6ms, and 250μs. The latter category can be further divided into three others based on bandwidth requirements: those with 173Mbps, 1075Mbps, and 2500Mbps. Taking these five categories into account and aiming to represent several real-world splits while minimizing complexity, we consider $\mathcal{F} = \{f_1, f_2, f_3, f_4, f_5\}$ as the set of RAN functions that can be virtualized at either an RU or CU. Table 2 describes the mapping between these RAN functions and LTE splits ($s2$ to $s6$). Split $s6$ represents nothing but the basic signal processing running at RUs, i.e., C-RAN. This is labelled as f_0, a function that always run at every RU. On the other end, we consider split $s1$ with no delay requirement, in which all five RAN functions run at RUs. This split is devised in order to represent the D-RAN setting, i.e., when there is no split.

	Run at RU	Run at CU	Equivalent split	Bandwidth req.	Delay req.
$s1$	f_1, f_2, f_3, f_4, f_5	None	D-RAN	150Mbps	None
$s2$	f_1, f_2, f_3, f_4	f_5	RRC-PDCP PDCP-RLC	151Mbps	30ms
$s3$	f_1, f_2, f_3	f_4, f_5	RLC-MAC Split MAC	151Mbps	6ms
$s4$	f_1, f_2	f_3, f_4, f_5	MAC-PHY PHY split I	173Mbps	250μs
$s5$	f_1	f_2, f_3, f_4, f_5	PHY split II PHY split III	1075Mbps	250μs
$s6$	None (other than f_0)	f_1, f_2, f_3, f_4, f_5	PHY split IIIb PHY split IV $\rightarrow$ C-RAN	2500Mbps	250μs

Table 2. RAN function distribution and the splits they represent. Bandwidth requirement is considering downlink and a demand of 150Mbps.

Base stations and transport nodes. We consider a RAN with a set $\mathcal{B} = \{b_0, \ldots, b_{|\mathcal{B}|}\}$ of B base stations (BSs). A BS can act either as a CU, if it is a centralization point, or an RU. Similar to [Garcia-Saavedra et al. 2018a], in this work, we consider only one CU and define $b_0 \in \mathcal{B}$ as the BS (actually, the cell site) that will host the CU. We also consider $\mathcal{N} = \mathcal{B} \setminus \{b_0\} = \{b_1, \ldots, b_{|\mathcal{N}|}\}$ the set of N BSs that will act as RUs and $\mathcal{M} = \{m_1, \ldots, m_{|\mathcal{M}|}\}$ the set of M transport nodes (routers, switches). Transport nodes cannot do any processing and are used only for traffic forwarding. Associated to each BS $b_i \in \mathcal{B}$ there is: 1) a cost α_{b_i} to use BS b_i, and 2) the RAN demand (Mbps) λ_{b_i} for BS b_i.

To represent the network of a mobile operator, we define the graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_0\} \cup \mathcal{B} \cup \mathcal{M}$ being the set of network nodes and $\mathcal{E} = \{e_{v_i,v_j},\ v_i, v_j \in \mathcal{V}\}$ representing the set of network links connecting the nodes. v_0 represents the core and it is the source/destination for all flows. The core is considered because D-RAN is a viable configuration in our model. In addition, each link $e_{v_i,v_j} \in \mathcal{E}$ has a capacity $c_{e_{v_i,v_j}}$ in Mbps and a delay $d_{e_{v_i,v_j}}$ in ms.

Routing. We consider that all traffic in the network has the core as its source (downlink) or destination (uplink). However, without loss of generality, we will only represent the downlink case in this work. We consider flows from the core to the CU (to model the backhaul), from the CU to each RU (to model the fronthaul and split requirements) and from the core to the RU (to enable D-RAN and to complete the backhaul modeling). We define $\mathcal{P}$ as the set of all paths from the core to all BS $b_n \in \mathcal{B}$ and from the CU b_0 to all RUs $b_n \in \mathcal{N}$. Given an RU $b_i \in \mathcal{N}$, we also define the following:

- $\mathcal{P}^1 \subseteq \mathcal{P}$, such as $\mathcal{P}^1$ is the set of all paths starting at the core v_0 and ending at CU b_0;
- $\mathcal{P}^2 \subseteq \mathcal{P}$, such as $\mathcal{P}^2 = \bigcup_{b_i \in \mathcal{N}} \mathcal{P}^2_{b_i}$, where $\mathcal{P}^2_{b_i}$ is the set of all paths starting at the core v_0 and ending at b_i;
- $\mathcal{P}^3 \subseteq \mathcal{P}$, such as $\mathcal{P}^3 = \bigcup_{b_i \in \mathcal{N}} \mathcal{P}^3_{b_i}$, where $\mathcal{P}^3_{b_i}$ is the set of all paths starting at CU b_0 and ending at b_i.

Problem statement. Given the set of nodes $\mathcal{V}$ of the network, the set $\mathcal{E}$ of links that connect them, and the vRAN demand for each RU $b_i \in \mathcal{N}$, our goal is to determine a split for each RU, while respecting bandwidth and delay constraints and maximizing the centralization level of the vRAN functions in the network. In the next section, we formalize and solve this problem as a Mixed Integer Linear Programming (MILP) model.

4. Problem Formulation and Solution

We define the set of decision variables $u^{f_n}_{b_i} = \{0, 1\}$ to represent the vRAN function allocation, so that $u^{f_n}_{b_i} = 1$, if BS $b_i \in \mathcal{B}$ implements function $f_n \in \mathcal{F}$; and $u^{f_n}_{b_i} = 0$ otherwise. We define the decision variable $r_p \in \mathbb{R}^+ \cup \{0\}$ to represent the flow passing through each path $p \in \mathcal{P}$. The objective of the model is to maximize the centralization level of the network. Thus, we define the following objective function:

$$\min_{u,r} \quad \alpha_{b_0} \sum_{b_i \in \mathcal{N}} \sum_{f_n \in \mathcal{F}} (1 - u^{f_n}_{b_i}) + \sum_{b_i \in \mathcal{N}} \alpha_{b_i} \sum_{f_n \in \mathcal{F}} u^{f_n}_{b_i}, \tag{1}$$

where α_{b_0} and α_{b_i} represent the cost to use, respectively, CU b_0 and RU $b_i, \forall b_i \in \mathcal{N}$. Note that α_{b_0} and α_{b_i} allow to control the goal of the model. For example, for any value of α_{b_0} and α_{b_i} as long as $\alpha_{b_0} < \alpha_{b_i}$, the model tries to centralize more functions.

Function chaining. The splits of the RAN functions must preserve the protocol stack order during the processing. This means that functions processing must be chained in a sequence that do not violate such arrangement. For example, as every RU runs f_0, which represents signal processing, to run f_2 at an RU, we also need to deploy f_1 on it. This stands for any function $f_n \in \mathcal{F}$, i.e, every time a function higher up in the stack is deployed at an RU, this RU needs to implement all functions lower down in the stack. This reasoning also applies to the CU, but taking the protocol order from top-down. We assure such function chaining at the RUs and the CU with the following set of constraints:

$$u^{f_{n-1}}_{b_0} \leq u^{f_n}_{b_0}, \quad 2 \leq n \leq |\mathcal{F}| \tag{2}$$

$$u^{f_n}_{b_i} \leq u^{f_{n-1}}_{b_i}, \quad 2 \leq n \leq |\mathcal{F}|, \quad \forall b_i \in \mathcal{N}. \tag{3}$$

In addition, we need to make sure that, for each CU-RU pair, at least one of them virtualizes each RAN function, that is:

$$u_{b_i}^{f_n} + u_{b_0}^{f_n} \geq 1, \quad 1 \leq n \leq |\mathcal{F}|, \quad \forall b_i \in \mathcal{N} \tag{4}$$

$$u_{b_i}^{f_n} \in \{0, 1\}, \quad 1 \leq n \leq |\mathcal{F}|, \quad \forall b_i \in \mathcal{B}. \tag{5}$$

Bandwidth requirements. Constraints are also needed to ensure that each split will meet its bandwidth requirement. We consider that flows can be fractional, as long as the total flow satisfies the demand.

We first analyze the core flow for each split. For each CU-RU pair, if any level of centralization is achieved (splits $s2$ to $s6$), then the demand for that RU, coming from the core v_0, must first pass through the CU b_0, and so that demand has to be computed in the total core-CU flow. The only case when centralization does not occur for an RU-CU pair is the D-RAN setting, which is also the only case when $u_{b_i}^{f_5} = 1$. Thus, to fulfill core-CU flow requirements, we define $R_1(u, \lambda) = \sum_{b_i \in \mathcal{B}} (\lambda_{b_i} - u_{b_i}^{f_5} \lambda_{b_i})$ and ensure that the sum of all flows passing through paths starting at the core v_0 and ending at the CU b_0 adds up to at least $R_1(u, \lambda)$, that is:

$$\sum_{p \in \mathcal{P}^1} r_p \geq R_1(u, \lambda). \tag{6}$$

Analyzing the flow for each core-RU pair, as previously stated, it only exists in the D-RAN setting ($u_{b_i}^{f_5} = 1$). Thus, to fulfill core-RU flow requirements, we define $R_2(u_{b_i}, \lambda_{b_i}) = u_{b_i}^{f_5} \lambda_{b_i}$, so that:

$$\sum_{p \in \mathcal{P}_{b_i}^2} r_p \geq R_2(u_{b_i}, \lambda_{b_i}), \quad \forall b_i \in \mathcal{B}. \tag{7}$$

Finally, analyzing the flow for each CU-RU pair, we observe that the demand for the RU is transformed according to the requirements of the chosen split. Considering a linear relation, we find that function $R_3(u_{b_i}, \lambda_{b_i}) = (u_{b_i}^{f_3} - u_{b_i}^{f_5})(\lambda_{b_i} + 1) + (1 - u_{b_i}^{f_3})173 + (1 - u_{b_i}^{f_2})902 + (1 - u_{b_i}^{f_1})1425$ proper represents the flow requirements for splits $s2$ to $s6$ [Small Cell Forum 2016]. Note that in the D-RAN setting, the flow requirement for the pair CU-RU is zero since there is no interaction between them, hence:

$$\sum_{p \in \mathcal{P}_{b_i}^3} r_p \geq R_3(u_{b_i}, \lambda_{b_i}), \quad \forall b_i \in \mathcal{B}. \tag{8}$$

Delay requirements. Since delay requirements for the splits in our model fall into three categories, in addition to the no delay restriction for the D-RAN setting, we define three new sets of paths according to the aggregate delay of their constituent links:

- $\mathcal{P}_{b_i}^A \subseteq \mathcal{P}^3$: set of all paths starting at CU b_0 and ending at RU b_i, with delay greater than 30ms;
- $\mathcal{P}_{b_i}^B \subseteq \mathcal{P}^3$: set of all paths starting at CU b_0 and ending at RU b_i, with delay greater than 6ms;

- $\mathcal{P}^C_{b_i} \subseteq \mathcal{P}^3$: set of all paths starting at CU b_0 and ending at RU b_i, with delay greater than 250μs.

As shown in Table 2, the delay requirements for splits $s1$, $s2$, and $s3$ are, respectively, none, 30ms, and 6ms. Splits $s4$, $s5$, and $s6$ all have delay requirements of 250μs. We have to make sure that the flows for the chosen splits do not travel through paths in which the aggregate delay exceeds the requirement. This is achieved by making such flows equal to zero. Consider D a very big number [1], we can then represent the delay requirements as:

$$\sum_{p \in \mathcal{P}^A_{b_i}} r_p \leq D(4 - u^{f_1}_{b_i} - u^{f_2}_{b_i} - u^{f_3}_{b_i} - u^{f_4}_{b_i} + u^{f_5}_{b_i}), \quad \forall b_i \in \mathcal{B} \tag{9}$$

$$\sum_{p \in \mathcal{P}^B_{b_i}} r_p \leq D(3 - u^{f_1}_{b_i} - u^{f_2}_{b_i} - u^{f_3}_{b_i} + u^{f_4}_{b_i} + u^{f_5}_{b_i}), \quad \forall b_i \in \mathcal{B} \tag{10}$$

$$\sum_{p \in \mathcal{P}^C_{b_i}} r_p \leq D(u^{f_3}_{b_i} + u^{f_4}_{b_i} + u^{f_5}_{b_i}), \quad \forall b_i \in \mathcal{B}. \tag{11}$$

Note that split $s2$ would make the right hand side of (9) equal to zero, by zeroing the flows travelling through paths with delay greater than 30ms. Similarly, split $s3$ would make the right hand side of (10) zero, by making zero the flows travelling through paths with delay greater than 6ms. The splits $s4$, $s5$, and $s6$ would do the same to constraint (11), but for flows with delay greater than 250μs. Also, note that $\mathcal{P}^A_{b_i} \subseteq \mathcal{P}^B_{b_i} \subseteq \mathcal{P}^C_{b_i}$.

Link capacity. Finally, we have to make sure that the capacity of the links in the crosshaul is not exceeded by the flows travelling through them. We consider flows to be non-negative. Let $I_p^{e_{v_i,v_j}} \in \{0,1\}$, $\forall e_{v_i,v_j} \in \mathcal{E}$, $\forall p \in \mathcal{P}$, such that $I_p^{e_{v_i,v_j}} = 1$, if path p has link e_{v_i,v_j}; and $I_p^{e_{v_i,v_j}} = 0$, otherwise. We ensure that the capacity of the links in the crosshaul is not exceeded by employing the following constraints:

$$\sum_{p \in \mathcal{P}} r_p I_p^{e_{v_i,v_j}} \leq c_{e_{v_i,v_j}}, \quad \forall e_{v_i,v_j} \in \mathcal{E} \tag{12}$$

$$r_p \in \mathbb{R}^+ \cup \{0\}, \quad \forall p \in \mathcal{P}. \tag{13}$$

We thus can summarize the formulation of the problem as:

Problem 1 (plasticRAN Design Problem – PRD)

$$\min_{u,\,r} \quad \alpha_{b_0} \sum_{b_i \in \mathcal{N}} \sum_{f_n \in \mathcal{F}} (1 - u^{f_n}_{b_i}) + \sum_{b_i \in \mathcal{N}} \alpha_{b_i} \sum_{f_n \in \mathcal{F}} u^{f_n}_{b_i} \tag{1}$$

subject to:

Constraints (2) – (13)

[1] In our problem, two times the number of RUs times 2500 is sufficient for D.

Solution. The problem of jointly defining splits and routing for vRANs is $\mathcal{NP}$-hard, as shown by [Garcia-Saavedra et al. 2018a]. To increase our model efficiency and employ it in larger networks, we apply Benders Decomposition [Geoffrion 1972]. This optimization technique consists in dividing the model into a master and slave problem (according to its variables), and solving the model as a two stage process. In our case, we leave the set of binary u variables in the master problem along with its constraints, and move the continuous r routing variables and their constraints to the slave problem. In each iteration, we first solve the master problem, and then proceed to solve the slave with the fixed splits produced by the master. If an infeasible solution is reached, we add cuts to the master problem, so that it does not generate that set of splits again. The master problem is then repeatedly solved until no more cuts can be generated or until its solution reaches a certain bound. This strategy allow us to solve the original problem without all its constraints at once, adding them as needed.

5. Performance Evaluation

In this section, we evaluate plasticRAN and compare the obtained results with FluidRAN [Garcia-Saavedra et al. 2018a], a state-of-the-art solution to the vRAN design problem. Given a network topology and its link and capacity constraints, FluidRAN achieves the maximum vRAN centralization by selecting the optimal split and routing path for each RU-CU pair. However, different from our model, FluidRAN assumes that the CU is always placed at the core, and does not take into account the flows between core-CU and core-RU. As a result, D-RAN is not a feasible solution in FluidRAN.

In order to compare the solutions generated by both models, we convert FluidRAN splits into plasticRAN ones. Such conversion is possible since FluidRAN assumes fewer splits than our model. Table 3 illustrates how we implement this conversion. The models are compared through a metric of centralization level, which consists in the proportion between the number of centralized functions achieved by the model and the total number of RAN functions.

FluidRAN split	plasticRAN split	4G/5G split
$x, y = 1$	$f_1, f_2, f_3, f_4 = 1$ $f_5 = 0$	RRC-PDCP PDCP-RLC
$x = 1$ $y = 0$	$f_1, f_2, f_3 = 1$ $f_4, f_5 = 0$	RLC-MAC Split MAC
$x, y = 0$	$f_1, f_2, f_3, f_4, f_5 = 0$	PHY split IIIb PHY split IV $\rightarrow$ C-RAN

Table 3. FluidRAN splits and their equivalent in plasticRAN and 4G/5G.

Since real-world data about RAN topology is hard to come by, we generate synthetic network topologies for our evaluation. Particularly, we use Waxman graphs as they are shown to properly describe backhaul topologies [Lessmann 2015]. To generate the Waxman graphs, we use the library networkx 2.2 [2] in Python with parameters $\alpha = 0.1$, $\beta = 0.4$, and $L = 70$. The graph generator places n nodes uniformly distributed in a rectangular area and connects them with probability $p = \beta exp(-d/\alpha L)$, where d is the distance between two nodes. We set the link capacity and delay for each edge according to their length and the profiles shown in Table 4.

[2] https://networkx.github.io/

Technology	Bandwidth (Gbps)	Prop. delay (μs)	Distance (km)
mmWave (60-80 GHz)	0.9, 1.25, 1.5, 2, 3, 4, 8	1-20	0.3-6
μWave (6-60GHz)	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.25, 1.5, 2	1-100	0.3-30
Copper (1000/10G/40GBASE-T)	1, 10, 40	0.05-0.5, 0.275, 0.15	0.001-0.1, 0.055, 0.03
SMF fiber @ 1310 nm (1000, 10G, 40G, 100GBASE-EX, LR, LR-4)	1, 10, 40, 100	1-200, 50, 50, 50	0.2-40, 10, 10, 10
SMF fiber @ 1550 nm (1000, 10G, 40G, 100GBASE-ZX, ER, ER-4)	1, 10, 40, 100	1-350, 200, 200, 200	0.2-70, 40, 40, 40
TbE (*under development)	200, 400	1-50	0.2-10

Table 4. Profiles of links for topology generation [Garcia-Saavedra et al. 2018b].

We run all experiments at a virtual machine (VM) created using Xen 4.11. The VM runs Debian 9 GNU/Linux and is configured with 16 vCPUs, 64GB RAM, and 40GB of virtual disk. The VM is hosted in a server with two Intel Xeon Silver 4114 processors running at 2.20GHz. We implemented both models using Python 2.7.12, docplex 2.8.125, and IBM CPLEX 12.8.0 as the solver.

5.1. Centralization level

The first set of experiments evaluates the centralization level achieved by each model. In these experiments, we consider topologies with 30, 40, and 50 nodes. For each size, we generate 10 different topologies, and for each topology, we position the RUs, CU, and core in 30 different configurations. In each configuration, we choose 40% of the nodes to be the RUs, one node to be the CU, and another to be the core. The rest are considered transport nodes. At total, we run 900 experiments. For each experiment, we first pre-compute all the paths between CU-RU, core-CU, and core-RU. Our model uses all pre-computed paths, while FluidRAN uses only the CU-RU ones. The demand for each RU is set to $\lambda_{b_i} = 150Mbps, \forall b_i \in \mathcal{N}$.

Table 5 shows the percentage of feasibility for each model. The increased number achieved by plasticRAN is due to the inclusion of the D-RAN setting, which is a split that requires the least bandwidth and has no delay requirements.

	Feasible solutions	Infeasible solutions	Percentage of feasibility
plasticRAN	361	539	40.11%
FluidRAN	228	672	25.33%

Table 5. Proportion of feasible solutions for each model.

Figure 2 shows the cumulative distribution function (CDF) of the centralization level achieved by both models in 900 experiments, with dots representing the average. We can see that, overall, plasticRAN achieves higher levels of centralization than FluidRAN. This happens because plasticRAN considers a higher number of splits. However, the smallest centralization level achieved by FluidRAN is higher than the one experimented by plasticRAN. This is because FluidRAN considers at least PDCP implemented at the CU, while plasticRAN takes D-RAN as a feasible solution. Nonetheless, since the percentage of feasibility of our model is 15% higher than the one of FluidRAN, we believe that including the D-RAN setting promotes a better tradeoff. In addition, the few

experiments where FluidRAN achieves higher centralization levels than plasticRAN, despite the fact that our model takes more splits into consideration, can be explained by the fact that in FluidRAN routing is simpler, as the model only consider CU-RU flows. plasticRAN, on the contrary, has to route CU-RU, core-CU and core-RU flows, leading to a higher chance of link saturation. This forces the model to choose splits with less consumed bandwidth but lower centralization level.

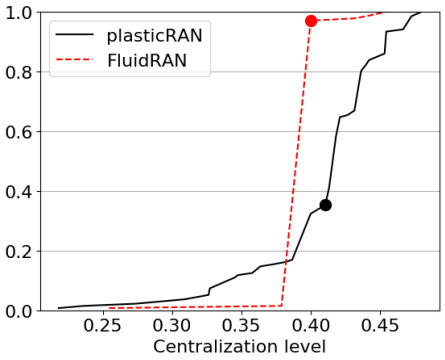

Figure 2. Centralization level for feasible solutions of 900 experiments.

Figure 3 shows the CDFs of the time spent to run each model as well as the CDF of the time taken to pre-compute the paths, considering topologies with 30, 40 and 50 nodes. Dots represent the average runtime. Since plasticRAN considers three types of flows (CU-RU, core-CU and core-RU) and FluidRAN only one (CU-RU), the execution time of our model is higher, as can be observed in the figure.

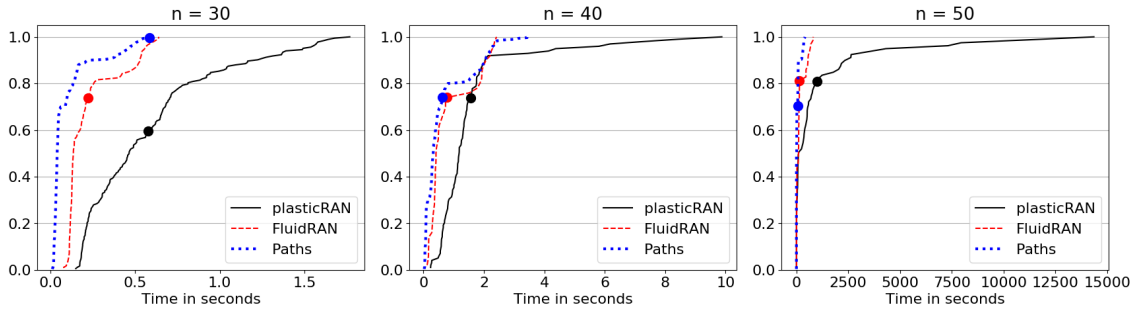

Figure 3. Execution time for topologies with size 30, 40 and 50.

In order to assess how the search space affects the routing complexity, Table 6 illustrates the mean number of paths processed by each model in each topology size. We can see that, although both models work with the same topologies, the mean number of processed paths in plasticRAN is higher. In addition, we can observe that, as the size of the topology increases slowly, the number of processed paths increases quickly. This happens even for FluidRAN. Indeed, during our experiments, we try to pre-compute the paths for topologies with 60 node. However, the problem showed intractable for both models. One way to overcome this problem is to generate less paths by pruning them based on depth. However, this would compromise the optimality of the solution.

To show the impact of different levels of pruning to both models, we redo the previous experiments discarding (in each model) all paths whose depth was higher than a

Mean of paths processed	n30	n40	n50
plasticRAN	591	1766	312960
FluidRAN	290	1320	160624

Table 6. Mean number of paths processed for different size topologies.

threshold (cutoff levels). Figure 4 shows the centralization level achieved by the models, when imposing cutoff levels of 7, 10, and 12. We can see that the centralization level does not fluctuate much between these cutoff levels. We also observe that the results are very similar to those obtained in Figure 2, in which no pruning is applied. Indeed, our experiments show that the centralization level is the same when applying no pruning or employing cutoff levels of 12 and 15 for topologies with 30, 40 and 50 nodes.

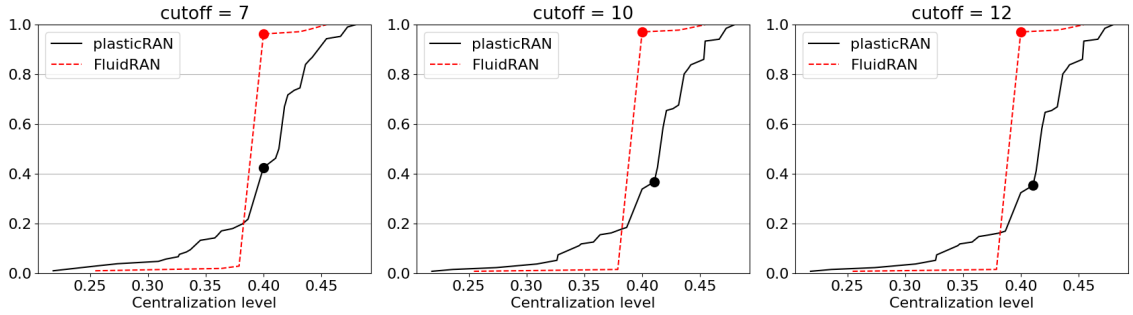

Figure 4. Centralization level when pruning paths with depth $\geq$ 7, 10, 12.

Despite this result, we can see in Table 7 that the mean number of processed paths decreases significantly when the different level of pruning is applied to the models, especially the lower ones (7,10). While such pruning reduces the execution time of the models, as shown in Figure 5, it also affects the number of feasible solutions. This is illustrated in Table 8. We can observe that plasticRAN is more affected by the pruning than FluidRAN, since our model has more paths to process. Thus, the chance of discarding a relevant path to achieve feasibility is higher in plasticRAN. In particular, a cutoff of 7 decreases in 17.45% the percentage of feasibility of plasticRAN. On the other hand, the same cutoff decreases in 6% the percentage of feasibility of FluidRAN. As illustrated in Table 8, in our experiments, cutoffs of 12 and 15 give the best trade-off between efficiency and feasibility for topologies up to 50 nodes. In summary, while pruning helps managing the number of variables instantiated in the models, it may affect their solutions. Since the number of processed paths tends to grow as the topology size increases, pruning paths based on depth will have a greater impact on feasibility when working with larger topologies.

5.2. CU positioning

One of the key decisions when planning a vRAN is the CU placement, since the CU position can increase or decrease the total centralization for the network. An ideal model for vRAN would include CU placement as a decision variable, but as we show in Section 5.1, even without including such variable the problem is already hard to solve. Usually, models that consider the CU at a fixed location make some assumptions. For example, FluidRAN [Garcia-Saavedra et al. 2018a] considers that the CU is always colocated with the

Mean of paths processed	n30	n40	n50
plasticRAN, cutoff 7	118	197	419
plasticRAN, cutoff 10	258	508	2469
plasticRAN, cutoff 12	392	786	6752
plasticRAN, cutoff 15	545	1244	25584
FluidRAN, cutoff 7	60	113	227
FluidRAN, cutoff 10	136	316	1432
FluidRAN, cutoff 12	204	532	4089
FluidRAN, cutoff 15	284	889	15333

Table 7. Mean number of paths processed considering different cutoffs.

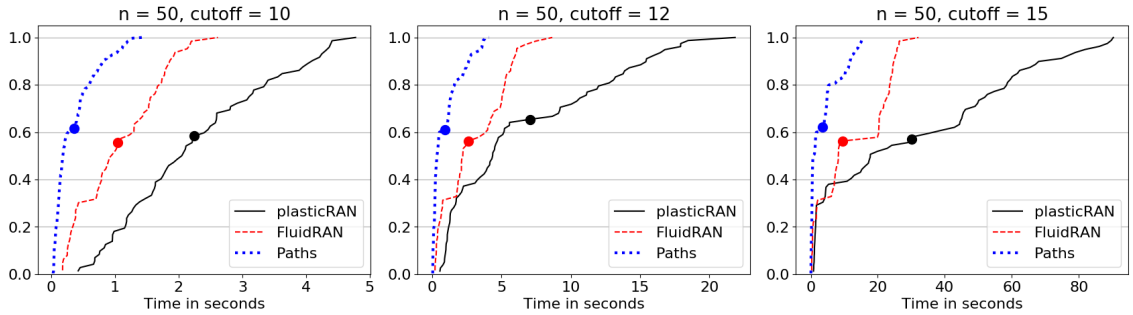

Figure 5. Execution time for topologies with 50 nodes and cutoff level of 10, 12, and 15.

core. In order to evaluate the impact of such assumption to the centralization level, we run experiments with 10 different topologies. For each topology, the RUs and the core are placed in fixed positions, while the CU placement varies as follows: 1) we run one experiment where the CU is colocated with the core; 2) for each transport node, we run one experiment where the transport node becomes the CU (thus, it is not colocated with the core). In our experiments, considering all the 10 different topologies we have tested, the best centralization level does not occur when the CU is collocated with the core. Figure 6 illustrates one of the tested topologies. We can see that the configuration on the left has CU and core colocated. However, the highest level of centralization is obtained using the configuration on the right, in which the mean path length between core and CU has 12.42 nodes.

	Feasible solutions	Infeasible solutions	Percentage feasible	Difference without cutoff
plasticRAN, cutoff 7	204	696	22.66%	17.45%
plasticRAN, cutoff 10	337	563	37.44%	2.67%
plasticRAN, cutoff 12	356	544	39.55%	0.56%
plasticRAN, cutoff 15	361	539	40.11%	No difference
FluidRAN, cutoff 7	174	726	19.33%	6%
FluidRAN, cutoff 10	226	674	25.11%	0.22%
FluidRAN, cutoff 12	228	672	25.33%	No difference
FluidRAN, cutoff 15	228	672	25.33%	No difference

Table 8. Proportion of feasible solutions considering different cutoff levels.

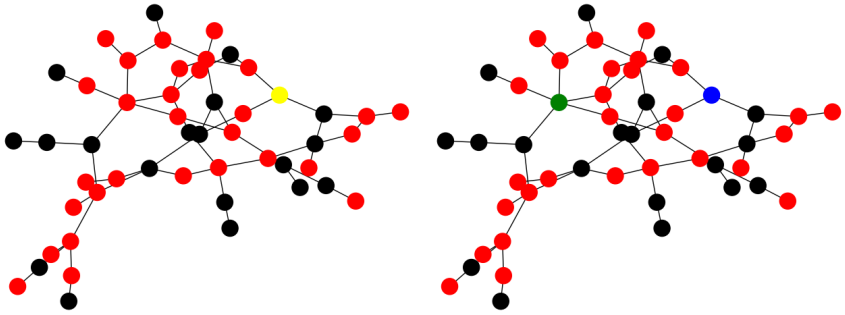

Figure 6. Left topology has CU colocated with core (yellow node). Right topology has CU (green node) in a different place than the core (blue node). Black nodes represent RUs and red nodes are transport nodes.

6. Conclusion

In this paper, we presented plasticRAN, a new optimization model for maximizing the vRAN centralization level. Our model takes into account jointly the functional splits and the routing over an integrated transport network, i.e., a crosshaul. We compared plasticRAN with a state-of-the-art solution and showed how our model improves the centralization level and reduces the number of infeasible solutions. Additionally, we evaluated the positioning of the CU in different places, including colocated with the core. In most scenarios, colocation is not the best choice.

As illustrated in the performance evaluation, the execution time of plasticRAN increases exponentially as function of the topology size. This happens because each path adds a decision variable to the model. While pruning contributes to limit the number of paths, it also increases infeasibility. The model demands the pre-computing of all simple paths, but the actual solution uses only a very small fraction of them. Thus, it would be also useful to deal with the paths only on-demand. Column Generation [Desaulniers et al. 2006], for example, is a potential method to apply for dealing with our routing subproblem.

Acknowledgement

The work is partially supported by the EU-BR project NECOS (777067).

References

Asensio, A., Saengudomlert, P., Ruiz, M., and Velasco, L. (2016). Study of the centralization level of optical network-supported Cloud RAN. In *2016 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6.

Chang, C., Nikaein, N., Knopp, R., Spyropoulos, T., and Kumar, S. S. (2017). FlexCRAN: A flexible functional split framework over ethernet fronthaul in Cloud-RAN. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7.

Chang, C., Nikaein, N., and Spyropoulos, T. (2016). Impact of Packetization and Scheduling on C-RAN Fronthaul Performance. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.

Checko, A., Avramova, A. P., Berger, M. S., and Christiansen, H. L. (2016). Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings. *Journal of Communications and Networks*, 18(2):162–172.

Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S., and Dittmann, L. (2015). Cloud RAN for Mobile Networks—A Technology Overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426.

Chen, X., Han, Z., Zhang, H., Xue, G., Xiao, Y., and Bennis, M. (2018). Wireless Resource Scheduling in Virtualized Radio Access Networks Using Stochastic Learning. *IEEE Transactions on Mobile Computing*, 17(4):961–974.

Costa-Perez, X., Garcia-Saavedra, A., Li, X., Deiss, T., de la Oliva, A., di Giglio, A., Iovanna, P., and Moored, A. (2017). 5G-Crosshaul: An SDN/NFV Integrated Fronthaul/Backhaul Transport Network Architecture. *IEEE Wireless Comm.*, 24(1):38–45.

de Lima, J. L. and Couto, R. S. (2018). Minimização da Latência no Posicionamento de Funções em Cloud RANs. *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 36:1–14.

de Souza, P. A., Abdallah, A. S., Bueno, E. F., and Cardoso, K. V. (2018). Virtualized Radio Access Networks: Centralization, Allocation, and Positioning of Resources. In *2018 IEEE International Conf. on Comm. Workshops (ICC Workshops)*, pages 1–6.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2006). *Column generation*, volume 5. Springer Science & Business Media.

Garcia-Saavedra, A., Costa-Perez, X., Leith, D. J., and Iosifidis, G. (2018a). FluidRAN: Optimized vRAN/MEC Orchestration. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 2366–2374.

Garcia-Saavedra, A., Salvat, J. X., Li, X., and Costa-Perez, X. (2018b). WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul. *IEEE Transactions on Mobile Computing*, 17(10):2452–2466.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260.

Lessmann, J. (2015). Resource optimization in realistic mobile backhaul networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 3861–3866.

Lin, Y., Shao, L., Zhu, Z., Wang, Q., and Sabhikhi, R. K. (2010). Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development*, 54(1):4:1–4:12.

Rost, P., Bernardos, C. J., Domenico, A. D., Girolamo, M. D., Lalam, M., Maeder, A., Sabella, D., and Wübben, D. (2014). Cloud technologies for flexible 5G radio access networks. *IEEE Communications Magazine*, 52(5):68–76.

Rost, P., Talarico, S., and Valenti, M. C. (2015). The Complexity–Rate Tradeoff of Centralized Radio Access Networks. *IEEE Transactions on Wireless Communications*, 14(11):6164–6176.

Small Cell Forum (2016). Small cell virtualization functional splits and use cases. Document 159.07.02, Release 7.

Suryaprakash, V., Rost, P., and Fettweis, G. (2015). Are Heterogeneous Cloud-Based Radio Access Networks Cost Effective? *IEEE Journal on Selected Areas in Communications*, 33(10):2239–2251.