Flex5G: Flexible Functional Split in 5G Networks

Davit Harutyunyan and Roberto Riggio
FBK CREATE-NET, Trento, Italy; Email: {d.harutyunyan,rriggio@fbk.eu}

***Abstract*—5G networks are expected to support various applications with diverse requirements in terms of latency, data rates and traffic volume. Cloud–RAN and densely deployed small cells are two of the tools at disposal of Mobile Network Operators to cope with such challenges. In order to mitigate the fronthaul requirements imposed by the Cloud–RAN architecture, several functional splits, each characterized by a different demarcation point between the centralized and the distributed units, have emerged. However, the selection of the appropriate centralization level (i.e., the functional split) still remains a challenging task, since a number of parameters have to be considered in order to make such a decision. In this paper, a virtual network embedding (VNE) algorithm is proposed to flexibly select the appropriate functional split for each small cell. The VNE is formulated as an Integer Linear Programming (ILP) problem whose objective is to jointly minimize the inter–cell interference and the fronthaul bandwidth utilization by dynamically selecting the appropriate functional split. Specifically, dynamic and static ILP–based algorithms are proposed. Finally, dynamic and static VNE heuristics are proposed to address the scalability problem of the ILP–based algorithms in case of dense and ultra–dense mobile networks, respectively.**

***Index Terms*—Virtual Network Embedding, Small Cells, Inter–cell Interference, C–RAN, Flexible Functional Split.**

I. Introduction

Compared to LTE and LTE–Advanced networks, 5G networks are expected to deliver a 1000 times increase in the system capacity, reduced round–trip delay and enhanced performance for cell–edge users. Many mobile network operators (MNOs) are using network densification as an efficient way to meet the aforementioned goals [1]. Albeit the usage of smaller cells has a number of advantages (e.g., decreased distance between nodes, reduced path loss and transmission power, higher frequency reuse factor), it poses also several challenges (e.g., increased total cost of ownership, increased power consumption, more frequent handovers, increased level of interference). By far, the most obvious downside of densely deployed small cells is that it dramatically increases the level of inter–cell interference, which may result in a significant performance degradation unless interference mitigation techniques are used.

Recent advances in Network Functions Virtualization (NFV) enabled MNOs to transit from the fully–decentralized RAN (D–RAN) architecture, where baseband processing and radio elements are co–located, to the fully–centralized Cloud–RAN (C–RAN) architecture [2], where baseband units are decoupled from the radio elements (termed Distributed Unit – DU) and consolidated in large datacenters (termed Centralized Unit – CU[1]). By decoupling baseband processing from the radio elements, C–RAN can lower the total cost of ownership for MNOs. The vaunted benefits of C–RAN are enhanced radio resource utilization and coordination across multiple cells. The drawbacks of C–RAN are the tight bandwidth and latency requirements imposed on the fronthaul (i.e. the links interconnecting CUs with DUs) where protocols like the common public radio interface (CPRI) [7] are used to carry the IQ samples over (typically) optical fibers.

The C–RAN and D–RAN architectures are two extreme concepts, both with advantages and disadvantages. In fact, while D–RAN requires relatively low backhaul capacity, it does not allow for joint signal processing. Conversely, C–RAN enables joint signal processing techniques, such as coordinated multi-point transmission (CoMP), at the price of higher backhaul requirements (e.g., bandwidth, latency). In order to tackle the aforementioned challenges, a number of intermediate functional splits, each characterized by a different demarcation point between DUs and CUs, have been proposed. Different criteria have to be considered in order to select the appropriate functional split. Following the current galloping pace in the mobile data traffic demand, it is our standpoint that implementing a fixed functional split is not a viable solution in the long run. Therefore, considering the mobile traffic demand and the daily traffic variations, the flexibility of dynamically choosing the optimal functional split is essential in order to efficiently employ the fronthaul bandwidth and baseband processing resources.

Flexible functional split allows reaping the benefits of different functional splits by enabling MNOs to flexibly change the split option based on their needs. For example, since difference functional splits are characterized by significantly different fronthaul bandwidth and latency requirements, MNOs can support specific QoS settings for each offered service (e.g., low latency, high throughput). Additionally, flexible functional split allows them to support specific user density and load demand in each geographical area.

In this paper, we formalize and solve a virtual network embedding problem (VNE) for 5G networks supporting different functional split options. We formulate the problem as an integer linear programming (ILP) problem in which virtual network requests are received from mobile virtual network operators (MVNOs) and are embedded by the infrastructure providers (InPs), having an objective of dynamically selecting the appropriate functional split option that can enable MNOs to jointly minimize the network–wide inter–cell interference *and* the fronthaul bandwidth utilization. In this work, we

[1]Notice that the 3GPP [3] terminology is used throughout the paper. However, other terminologies such as Remote Radio Head (RRH) and BaseBand Unit pool (BBU pool), Remote Radio Unit (RRU) and Radio Cloud Center (RCC), and Radio Unit (RU) and Digital Unit (DU) for the DU and CU can be found in the technical documents of, respectively, SCF [4], NGFI [5] and NGMN [6].

Research leading to these results received funding from the European Unions H2020 Research and Innovation Action under Grant Agreement H2020-ICT-2014-2/671639 (COHERENT Project).

extend our previous study [8] by (i) proposing dynamic and static ILP–based algorithms, and scalable dynamic and static heuristics to solve the VNE problem, (ii) discussing the pros and cons and the applicability of each algorithm to different scenarios, and (iii) apart from the star substrate topology studied in the original work, discussing the possibility of applying the flexible functional split to a ring and a tree fronthaul topologies, highlighting their additional benefits.

The rest of this paper is structured as follows. The related work is discussed in Sec. II. The functional splits considered in this work are introduced in Sec. III. The substrate network and the virtual network request models are detailed in Sec. IV. The problem formulation and the algorithms are presented in Sec. V. The numerical results are reported in Sec. VI followed by Sec. VII comparing the PHY–RF split with the considered flexible functional split. Finally, Sec. VIII draws the conclusions.

II. Related Work

CU Placement. A sizeable body of work has been published on the CU placement problem [9], [10], [11], [12], [13]. In [9], the authors propose a Colony–RAN architecture for cellular systems, which is able to change the cell layout by dynamically adapting the connections between CUs and DUs according to the network conditions. An optimization algorithm is presented in [10] for the CU Placement problem over Fixed/Mobile Converged optical networks. The authors formulate an ILP problem, which efficiently calculates the minimum number of required CUs taking into account the maximum allowed distance between DUs and their CUs. The same authors propose an energy–efficient CU Placement algorithm in optical networks in [11], aiming at minimizing the Aggregation Infrastructure Power. An ILP optimization problem is formalized in [12] for optimizing cells assignment to different CUs. Statistical multiplexing gain and required fiber length are used as key performance indicators. An analytical model is derived in [13] to find the optimal ratio between optical fiber and microwave links in the fronthaul of mobile networks.

In [14], C–RAN is compared to the traditional D–RAN in terms of cost and energy consumption. A two–stage design, namely downlink OFDMA resource allocation and power allocation, and DU–CU assignment for a dynamic resource allocation, mechanism is proposed in [15]. In the first stage, an MILP problem is formulated for finding the best DU assignments and the best PRB allocations for the mobile users considering the SINR threshold for each of them. In the second stage, by employing the results obtained in the first stage, a DU–CU assignment problem is formulated as a Multiple Knapsack Problem (MKP), taking into account the real–time traffic load in DUs. The authors of [16] study a CU deployment problem considering the PHY–RF split and an intra–PHY–layer split, and their embedding problems are formalized and solved using an exact algorithm and a heuristic. The objective function aims at minimizing the total deployment cost of the mobile network while satisfying the traffic demands.

Network Sharing. Network sharing has been evolving since the arrival of 3G networks, reaching from passive sharing such as sharing of site locations, antenna masts, to active sharing techniques such as multi–operator core network (MOCN) and gateway core network (GWCN). The active network sharing has paved a way for new business opportunities, enabling InPs to host MVNOs, over–the–top (OTT) service providers and vertical market players over their physical network. The authors of [17] introduce an on–demand capacity broker concept, which is able to securely expose selected service features via APIs, allowing InPs to allocate the required portion of their networks to MVNOs, OTTs, or vertical market players. In [18], the authors present a VNF placement problem in which InP's RAN can be shared among several MVNOs.

Being inspired by the concept of "everything" as a service (XaaS) and having the goal of allowing mobile network operators to offer a customizable end–to–end service to MVNOs, the Network Slice–as–a–Service (NSaaS) concept is introduced in [19]. More specifically, three service models are proposed: application level, network function level and infrastructure level. Additionally, several implementation possibilities such as network slicing only in the core network, only in the RAN or in both in the core network and in the RAN, are discussed. The last slicing option (slicing in both core network and RAN), which implies that each slice of the RAN is connected to a core slice, along with the infrastructure level service model is of interest for our work. In the infrastructure level service model, network resources such as PRBs, antennas, DUs, CPU cores, memory and storage can be allocated to each of the MVNOs, guaranteeing resource isolation between them.

An extensive survey on network slicing can be found in [20], highlighting the major problems associated with network slicing and isolation between slices. Various approaches of wireless slicing are presented for different technologies such as LTE, WiMAX and Wi–Fi. Whereas, a detailed study on the impact of network slicing in 5G RANs can be found in [21].

Flexible Functional Split. The functional split problem has attracted a significant attention from both the academia and the industry [5], [6], [3]. There are in fact different approaches to small cell virtualization in terms of the point at which base stations operations are decomposed into physical and virtual. A number of factors (e.g., mobile data traffic demand, inter–cell interference scenario and latency constraints) have to be considered in order to select the optimal split point.

A detailed discussion on various functional splits can be found in [22], [23], [24], [25]. The authors of [22] propose a novel RAN–as–a–Service (RANaaS) concept in which centralization of management and processing is flexible and can be adapted to the actual service demands. Several functional splits are introduced, and numerical results on the required backhaul data rates for each envisioned split case are provided in [23]. While the authors of [24], [25] conduct a detailed investigation on various PHY–layer functional splits.

The authors of [26] propose a graph–based algorithm for analyzing different baseband functional splits. Several wired/wireless transport fronthauling technologies as well as the associated bandwidth and latency requirements for different functional splits are explored in [27]. A case–study analysis is presented in [28] for several PHY–layer functional splits, considering a digital subscriber line, microwave and optical fiber transport as fronthaul technologies. The authors conclude that among the different functional split, the PHY–RF split with optical fiber fronthaul option is the most profitable one, though it incurs the highest deployment costs. Based on burstiness of the traffic and the fact that the mobile data traffic varies depending upon the area (e.g., residential, office)

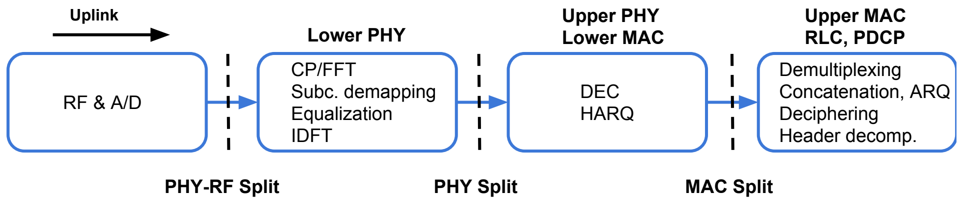


Fig. 1: Signal processing along with some of the functional split options within the RAN protocol stack in LTE networks.

TABLE I: Bandwidth and one–way latency requirements (absolute and relative) for different functional splits.

Splits	DL bandwidth	Latency	Latency class
PHY-RF Split	2.46 Gbps ($x1$)	250 μs ($x1$)	Ideal
PHY Split	0.93 Gbps ($x2.5$)	2 ms ($x8$)	Near Ideal
MAC Split	0.15 Gbps ($x16.5$)	6 ms ($x24$)	Sub Ideal

and the time of a day, mathematical and simulation methods are proposed in [29] for quantifying the multiplexing gain of the PHY–RF and the PDCP/RLC functional splits. In [30], a theoretical study is presented that jointly considers functional split selection and scheduling policy. Optimization problems are formulated aiming at minimizing the total latency, which is computed as the sum of the scheduling latency, the processing latency at the DUs and the CU pool, and the fronthaul transmission latency over all DUs. Different from the mentioned works, a control plane / user plane split is discussed in [31], highlighting the pros and cons.

However, to the best of our knowledge, this is the first work that considers the possibility of dynamically adapting a small cell functional split based on the evaluated inter–cell interference level at each small cell.

III. Functional Splits

In this section, we introduce the functional splits that are considered in this work. Figure 1 illustrates the basic signal processing blocks of the LTE stack in the uplink direction, highlighting the points at which a split is possible. The splits considered in this work are symmetrical for the uplink and for the downlink. Table I compares the functional splits in terms of fronthaul bandwidth and latency requirements [4].

PHY–RF Split. The PHY–RF split corresponds to full resource centralization with all baseband signal processing taking place at the CU pool, leaving the RF functions (e.g., analogue–to–digital and reverse conversion, signal amplification) at the DU side. While this functional split provides several advantages in terms of energy efficiency, computational diversity and improved spectral efficiency [2], its tight requirements in term of fronthaul bandwidth and latency can undermine its economical convenience.

PHY Split. By placing some of the physical layer functionalities such as FFT/IFFT, subcarrier mapping/demapping, signal equalization and MIMO processing at the DUs, it is possible to significantly relax the fronthaul requirements in terms of both bandwidth and latency. As it can be seen in Table I, taking the requirements of the PHY–RF split as a baseline, the PHY split allows the fronthaul bandwidth requirements to be reduced by a factor of 2.5. This is due to the removal of the cyclic prefix from the baseband signal and due to the fact that only received signals of the allocated PRBs are forwarded to the CU pool, therefore, providing a statistical multiplexing gain. Similarly, the fronthaul latency requirements are also relaxed by a factor of 8 when the PHY split is used. Notice however how these requirements are relieved at the expense of reduced resource centralization gain. For example, compared to the PHY–RF split, CoMP features such as joint transmission/reception can no longer be employed with the PHY split [32]. This can result in lower performances especially for cell–edge users, which are the ones that benefit the most from the interference reduction/cancellation features enabled by CoMP.

MAC Split. In this case, the HARQ procedure is taking place at the DU while the rest of the MAC functions along with the upper layers are consolidated at the CU pool. Compared to the PHY–RF split, the MAC split allows relaxing the latency requirements by a factor of 24, and the bandwidth requirements by a factor of 16.5. Functions such as joint decoding can no longer be exploited while joint scheduling and joint path selection are still possible.

IV. Network Model

This section details the substrate and the virtual network models. Figure 2 depicts the reference network architecture used in this work. The main idea of this figure is to show that different functional splits can co–exist at the same network and can be changed dynamically. In the lower left part of the figure we can see the traditional D–RAN architecture in which the DU and the CU are deployed in close proximity. As opposed to the D–RAN case, for all the other functional splits, the DUs are decoupled from the CUs, and an optical[2] fronthaul is used for their interconnection. It is important to say, regardless of the split option being employed at the given time, apart from the CU pool, also all the DUs possess processing capabilities since we consider a network in which the functional split options can be flexibly changed. For example, although in the case of employing the PHY–RF split there is no baseband signal processing at the DUs, the DUs are still required to have baseband signal processing capabilities in order to be able to support the other splits if necessary.

A. Substrate Network Model

Let $G_s = (N_s, E_s)$ be an *undirected* graph modeling the physical network, where $N_s = N_s^1 \cup N_s^2$ is the set of $n_1 = |N_s^1|$ DUs and $n_2 = |N_s^2|$ CU pools, and $E_s \subseteq N_s^1 \times N_s^2$ is the set of fronthaul links. An edge $e^{nm} \in E_s$ if and only if a connection exists between $n, m \in N_s$. Three weights, $\omega^s_{ant}(n)$, $\omega^s_{prb}(n)$ and $\omega^s_{prc}(n)$, are assigned to each node $n \in N_s : \omega^s_{ant,prb,prc}(n) \in \mathbb{N}^+$ representing, respectively, the number of RF front–ends, the set of physical resource blocks

[2]Optical fiber, as the most common fronthauling option, has been selected as the fronthaul medium. However, other fronthauling options such as millimeter wave wireless links or copper links are also possible [27].

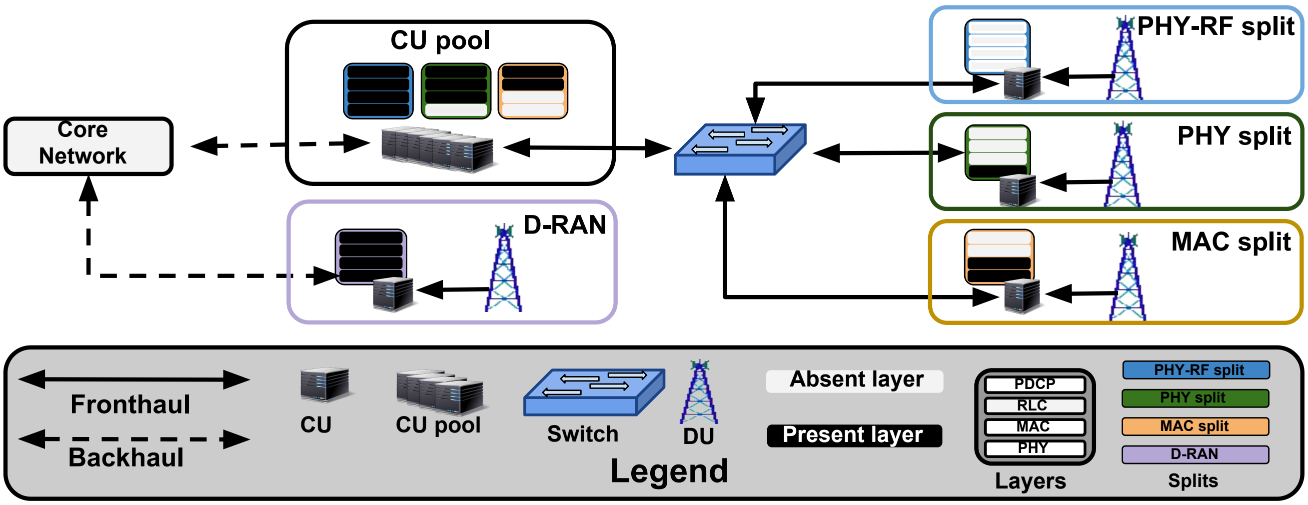


Fig. 2: This figure shows the coexistence of different functional splits at the CU pool. Notice that, apart from the CU pool, also the DUs possess processing capabilities, regardless of the functional split option being employed at the considered time.

TABLE II: Substrate network parameters

Variable	Description
G_s	Substrate network graph.
N_s	Substrate nodes in G_s.
N_s^1	Substrate DUs in G_s.
N_s^2	Substrate CU pools in G_s.
E_s	Substrate links in G_s.
$\omega^s_{ant}(n)$	Number of RF front–ends available at the node $n \in N_s$.
$\omega^s_{prb}(n)$	Set of PRBs available at the node $n \in N_s$.
$\omega^s_{prc}(n)$	The processing capacities of the node $n \in N_s$.
$\omega^s_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_s$ (in Gbps).
$loc(n)$	Geographical location of the node $n \in N_s$.
$\delta(n)$	Coverage radius of the node $n \in N_s$ (in meters).
$R_n^{n'}(m)$	Fronthaul bandwidth needed on the link of node $n \in N_s$ that uses the m^{th} split to host virtual node $n' \in N_v$ (in Gbps).
Λ_{bwt}	Cost for each Gbps of bandwidth resource.
Λ_{itf}	Cost for each interfering PRB.

(PRBs) whose cardinality depends on the employed carrier, and the processing capacity supported by the node. Each substrate node is also associated with a geographic location $loc(n)$, as x, y coordinates, and a coverage radius $\delta(n)$, in meters, indicating the coverage area of the small cell centered on the node $n \in N_s$. Another weight $\omega^s_{bwt}(e^{nm})$ is assigned to each link $e^{nm} \in E_s$: $\omega^s_{bwt}(e^{nm}) \in \mathbb{N}^+$ representing the capacity (in Gbps) of the link connecting the two nodes. Table II summarizes the substrate network parameters.

Notice how, it is our assumption that the CU pool is equipped with enough computational capacity to support all the DUs employing the lowest possible functional split i.e., the PHY–RF split, which requires all baseband signal processing to take place at the CU pool. Whereas, the DUs are equipped with enough computational capacity to process the signals with the highest possible functional split, i.e. the MAC split. Additionally, we assume that the fronthaul links have enough capacity to support the PHY–RF split and that only DUs are equipped with RF front–ends [4].

B. Virtual Network Request Model

There are different approaches to model virtual network requests, from resource–based [33] [34] models to throughput–based models [35]. In this work, we use a resource–based model in which MVNOs can request one or more small cells with a particular antenna configuration and a fixed amount of PRBs to be allocated to their small cells. This model does not provide any throughput guarantees to the MVNO's users whose performances can be affected by users distribution and by the time–varying nature of the wireless channel.

Virtual network requests are modeled as *undirected* graphs $G_v = (N_v, E_v)$ where $N_v = N_v^1 \cup N_v^2$ is the set of $n_1 = |N_v^1|$ DUs and $n_2 = |N_v^2|$ CU pools, and $E_v \subseteq N_v^1 \times N_v^2$ is the set of virtual fronthaul links. Each node $n \in N_v$ in the virtual network requests has two weights, $\omega^v_{ant}(n)$ and $\omega^v_{prb}(n)$ indicating, respectively, the number of RF front–ends and the number of PRBs, while each DU $n \in N_v^1$ is also associated with a geographic location $loc(n)$, as x, y coordinates. This information together with the substrate DU location and its coverage radius is used to express how far a virtual DU $n \in N_v^1$ can be placed from the preferred location specified by $loc(n)$. Table III summarizes the virtual network request parameters.

Notice that MVNOs do not request processing resource at the CU pools or DUs. Neither do they request fronthaul bandwidth. Given the chosen functional split of a virtual small cell, which is the one of the host substrate small cell, and considering its requirements ($\omega^v_{ant}(n)$, $\omega^v_{prb}(n)$), the fronthaul bandwidth required to support the given virtual small cell can easily be derived [4]. For example, if the virtual DU $n \in N_v^1$ requests $\omega^v_{ant}(n) = 2$ (2×2 MIMO configuration) RF front–ends and $\omega^v_{prb}(n) = 50$ PRBs, this translates to the fronthaul bandwidth of $\omega^v_{bwt} = 2.46$ Gpbs, $\omega^v_{bwt} = 0.54$ Gpbs and $\omega^v_{bwt} = 0.046$ Gbps, respectively, in the cases of the PHY–RF split, the PHY split and the MAC split, assuming the transport block size of 21384 bits in an LTE network with 20 MHz bandwidth. Notice that as opposed to the PHY split and the MAC split, the fronthaul bandwidth for the PHY–RF split does not depend on the requested number of PRBs.

V. Virtual Network Embedding

In this section, the virtual network embedding problem is introduced followed by the dynamic and static ILP–based algorithms (*ILP–DM* and *ILP–ST*), and by the scalable dynamic and static embedding heuristics (*HEU–DM* and *HEU–ST*).

TABLE III: Virtual network request parameters

Variable	Description
G_v	Virtual network request graph.
N_v	Virtual nodes in G_v.
N_v^1	Virtual DUs in G_v.
N_v^2	Virtual CU pools in G_v.
E_v	Virtual links in G_v.
$\omega_{ant}^v(n)$	Requested number of RF front–ends at the node $n \in N_v$.
$\omega_{prb}^v(n)$	Requested number of PRBs at the node $n \in N_v$.
$\omega_{prc}^v(n)$	Processing capacity required at the node $n \in N_v$.
$\omega_{bwt}^v(e^{nm})$	Capacity required for the link $e^{nm} \in E_v$ (in Gbps).
$loc(n)$	Desired geographical location for the DU $n \in N_v^1$.

A. Problem Statement

The virtual network embedding problem studied in this work can be formally stated as follows:

Given: a star–shaped substrate topology, similar to the one depicted in Fig. 7a, where each node has its geographical location, processing capacity, fronthaul link capacity, RF front–ends and bandwidth expressed in terms a number of PRBs.

Find: the PRB allocations for each virtual node in a virtual network request, the functional split option employed at each substrate node and the fronthual link bandwidth allocation.

Objective: minimize the level of inter–cell interference and select the optimal functional split option for each substrate node in such a way as to also minimize the fronthaul bandwidth required to embed the virtual network request and to suppress the inter–cell interference level by using techniques/algorithms enabled by the selected functional split.

Upon arrival of a new virtual network request, the substrate network must decide whether it is to be accepted or rejected. The embedding process consists of two steps: node embedding and link embedding. In the node embedding step, each virtual node in the request is mapped to a substrate node. In the link embedding step, each virtual link is mapped to a single substrate path. In both cases, some constraints must be satisfied.

B. Dynamic ILP–based placement algorithm (ILP–DM)

In this subsection, the *ILP–DM* algorithm is presented. *ILP–DM* employs a dynamic embedding strategy, meaning that with the arrival of a new virtual network request, the request along with the ones that have been previously embedded are re–embedded. Thus, with every embedding, the optimal embedding solution is found for all the requests.

Every virtual DU $n' \in N_v^1$ in a request has a desired location $loc(n')$. Whereas, every substrate DU $n \in N_s^1$ has both a location $loc(n)$ and a coverage radius $\delta(n)$. For each virtual DU $n' \in N_v^1$, we can then define a cluster of candidate DUs $\Omega(n')$ to which the virtual DU $n' \in N_v$ can be mapped:

$$\Omega(n') = \Big\{ n \in N_s^1 | dis(loc(n), loc(n')) \leq \delta(n) \Big\} \qquad (1)$$

We can now provide the ILP formulation for our VNE problem. The objective of this formulation is to minimize the inter–cell interference at each small cell and, at the same time, minimize the fronthaul bandwidth required to serve the virtual network requests. The chosen objective function is:

$$min\Big(\sum_{n' \in N_v^1} \sum_{n \in \Omega(n')} \sum_{n^\star \in \Omega(n)} \sum_{p \in \omega_{prb}^s(n), p^\star \in \omega_{prb}^s(n^\star)}^{p=p^\star} \Lambda_{itf} \Phi_p^n \Phi_{p^\star}^{n^\star} + \sum_{n \in N_s^1} \sum_{n' \in N_v^1} \sum_{m \in \{1,2,\ldots,|R_n^{n'}|\}} \Lambda_{bwt} R_n^{n'}(m) \Phi_{n,m}^{n'} \Big) \qquad (2)$$

where (with a slight abuse of notation) we use $n^\star \in \Omega(n)$ to indicate that the node $n^\star \in N_s^1 (n^\star \neq n)$ has overlapping radio coverage with the candidate substrate node $n \in \Omega(n')$ (i.e. an interfering node). Table IV summarizes all binary decision variable used in the ILP problem formulations.

The first term in the objective function aims at minimizing the number of overlapping PRBs at each host small cell, while the second term minimizes the fronthaul bandwidth requirements by taking into account the first term and selecting the most optimal functional split for each host small cell. The rationale behind this approach is that different functional splits can enable different interference management techniques [32], and thus a trade–off exists between fronthaul bandwidth requirements and the level of acceptable interference in the system. It is important to mention that we select the cost of each Gbps of bandwidth resource large enough from the cost of each interfering PRB ($\Lambda_{itf} \ll \Lambda_{bwt}$) in order to make sure that second term in the objective function is significantly larger compared to the first term. Nevertheless, the first term, although negligible, has to be present in the objective function in order to achieve the multi–objective optimization; that is, the joint minimization of network–wide inter–cell interference and fronthaul bandwidth requirements. This is because in the considered scenario the InP aims (i) to avoid allocating overlapping PRBs to the virtual small cells that have been hosted by neighbor small cells, and (ii) once the overlapping PRB allocation becomes inevitable, considering the level of inter–cell interference, to select the optimal functional split option for each host small cell with the goal of minimizing the fronthaul bandwidth requirements and, thanks to the selected functional split, being able to employ inter–cell interference management techniques aiming to reduce the interference.

In order to minimize the inter–cell interference, we first need to quantify it. For each small cell, a collision domain is selected, containing all the small cells with which the considered small cell has an overlap in its coverage area. In essence, these would be the small cells whose signals in the downlink may interfere with the downlink signals of the considered small cells. The inter–cell interference[3] can then be estimated based on the information of the PRB chunks allocated to virtual network requests (i.e., MVNOs).

Note that the objective function contains a quadratic term $\Phi_p^n \Phi_{p^\star}^{n^\star}$ that results in a standard (non–convex) quadratic formulation. To linearize this term, we define a variable $\Phi_{p,p^\star}^{n,n^\star}$ and substitute it to the quadratic term in the objective function:

$$\Phi_p^n \Phi_{p^\star}^{n^\star} \approx \Phi_{p,p^\star}^{n,n^\star} = \begin{cases} 1 & if \quad \Phi_p^n = \Phi_{p^\star}^{n^\star} = 1 \\ 0 & otherwise \end{cases} \qquad (3)$$

[3]In this study, the interference estimation is based on the worst case assumption; that is, the PRBs allocated to MVNOs are always in use. We do not consider the wireless users, and therefore, do not model their channels. More accurate interference estimation is, however, possible taking into account the PRB allocation of each user and their channel quality information.

TABLE IV: Binary decision variables $\{0, 1\}$

Variable	Description
$\Phi_p^n, \Phi_{p^\star}^{n^\star}$	Show if the PRBs $p, p^\star$ are in use at the substrate nodes n and $n^\star$, respectively.
$\Phi_{p,p^\star}^{n,n^\star}$	Shows the presence of inter–cell interference at $n^\star \in \Omega(n)$ and at $n \in \Omega(n')$ substrate DUs.
$\Phi_n^{n'}$	Shows if the virtual DU $n' \in N_v^1$ has been mapped to the substrate node $n \in \Omega(n')$.
$\Phi_{n,m}^{n'}$	Shows if the virtual DU $n' \in N_v^1$ has been mapped to the m^{th} functional split option of the substrate DU $n \in \Omega(n')$.
$\Phi_{n,p}^{n'}$	Shows if the PRB $p \in \omega_{prb}^s(n)$ of the substrate DU $n \in N_s^1$ has been allocated to the virtual DU $n' \in N_v^1$.

We will now detail the constraints used in the ILP formulation. The following constraint ensures that if the same PRB is being used by two or more small cells that are in the same collision domain then a penalty is applied to each of the small cell that uses the PRB:

$$\Phi_p^n + \Phi_{p^\star}^{n^\star} - \Phi_{p,p^\star}^{n,n^\star} \leq 1 \tag{4}$$

$$\forall n^\star \in \Omega(n), \quad \forall n \in \Omega(n'), \quad \forall n' \in N_v^1,$$

$$\forall p = p^\star \;\; p \in \omega_{prb}^s(n), p^\star \in \omega_{prb}^s(n^\star)$$

In essence, $\Phi_{p,p^\star}^{n,n^\star} = 1$ indicates the presence of the inter–cell interference at $n^\star \in \Omega(n)$ and $n \in \Omega(n')$ substrate DUs. The penalty (i.e., inter–cell interference) increases with an increase in the number of overlapping PRBs used by the small cells belonging to the same collision domain. The following constraints deal with, respectively, the required number of RF front–end and PRB resources, making sure that they are at most equal to the resources available at the substrate nodes:

$$\sum_{n' \in N_v^1} \omega_{ant}^v(n')\Phi_n^{n'} \leq \omega_{ant}^s(n) \quad \forall n \in N_s^1 \tag{5}$$

$$\sum_{n' \in N_v^1} \omega_{prb}^v(n')\Phi_n^{n'} \leq |\omega_{prb}^s(n)| \quad \forall n \in N_s^1 \tag{6}$$

It is worthwhile to note that the processing resources at the DUs/CU pools as well as the fronthaul bandwidth resources are not requested by MVNOs. Those resources depend on the actual functional split of the host substrate small cells and are computed by the InPs after having the optimal mapping solution of the virtual network requests.

Each requested virtual node $n' \in N_v$ must be mapped only once (7), while each virtual DU must be mapped only on a substrate DU that belongs to its cluster of candidates (8):

$$\sum_{n \in N_s} \Phi_n^{n'} = 1 \quad \forall n' \in N_v \tag{7}$$

$$\sum_{n \in N_s^1 \setminus \Omega(n')} \Phi_n^{n'} = 0 \quad \forall n' \in N_v^1 \tag{8}$$

The next constraint prevents the re–allocation of PRBs, making sure that each PRB is allocated maximum once:

$$\sum_{n' \in N_v^1} \Phi_{n,p}^{n'} \leq 1 \quad \forall n \in N_s^1, \quad \forall p \in \omega_{prb}^s(n) \tag{9}$$

Virtual DU embedding and PRB allocation must be consistent, meaning that if a virtual DU has been mapped to a given substrate DU then only the PRBs of that substrate DU must be allocated to the virtual DU:

$$\sum_{p \in \omega_{prb}^s(n)} \Phi_{n,p}^{n'} - \omega_{prb}^v(n')\Phi_n^{n'} = 0 \quad \forall n' \in N_v^1, \quad \forall n \in \Omega(n') \tag{10}$$

In order to compute the fronthaul bandwidth requirement for the virtual DU $n' \in N_v^1$, it has to be mapped to the functional split of the host substrate DU $n \in \Omega(n')$:

$$\Phi_n^{n'} - \sum_{m \in \{1,2,\ldots,|R_n^{n'}|\}} \Phi_{n,m}^{n'} = 0 \quad \forall n' \in N_v^1, \quad \forall n \in \Omega(n') \tag{11}$$

Finally, the last constraint handles the functional split selection for each substrate small cell ensuring that all the virtual small cells that have been mapped to the same substrate small cells have selected the same single functional split of the substrate small cell:

$$\sum_{n^\star \in \Omega(n)} \sum_{p \in \omega_{prb}^s(n), p^\star \in \omega_{prb}^s(n^\star)}^{p=p^\star} \Phi_{p,p^\star}^{n,n^\star} \leq I(m) \sum_{n^\star \in \Omega(n)} |\omega_{prb}^s(n^\star)|$$

$$\forall n' \in N_v^1, \quad \forall n \in \Omega(n'), \quad \forall m \in \{1, 2, ..., |R_n^{n'}|\} \tag{12}$$

where $I(m)$ represents the acceptable inter–cell interference level for each m^{th} functional split option (see Table V). This constraint effectively puts an upper bound on the number of acceptable overlapping PRB allocations. For example, for a PHY–RF split we are willing to accept as many PRB allocation overlaps as the number of PRBs in a collision domain. This essentially results in a reuse factor of 1, which is acceptable since the PHY–RF split enables several advanced interference mitigation techniques to be employed. Conversely, as the functional split moves up in the protocol stack we reduce the maximum number of allowed overlaps in the PRB allocations.

C. Static ILP–based placement algorithm (ILP–ST)

The *ILP–ST* algorithm resembles the *ILP–DM* algorithm in terms of the fact that both have the same objective function and that all the constraints defined for the *ILP–DM* algorithm are held also for the *ILP–ST* algorithm. However, the difference between them is that, in the case of the *ILP–ST* algorithm, as opposed to the *ILP–DM* one, virtual network requests are embedded sequentially. In other words, with the arrival of a new virtual network request, only that request is embedded. Thus, as the name of the algorithm implies, a static embedding is considered. As we will see in Sec. VI, this algorithm can be used to solve larger embedding problems since the static embedding is significantly faster compared to its dynamic counterpart. It is important to mention that in dynamic placement scenarios, G_v represents the composition of all the virtual network graphs, including the one of the new virtual network request, that are to be mapped with the arrival of a new virtual network request. Whereas in the static embedding scenarios, G_v represents the virtual network graph that characterizes only the new virtual network request.

D. Scalable dynamic placement heuristic (HEU–DM)

The *ILP–DM* VNE algorithm becomes computationally intractable as the size of the substrate network and/or of the virtual network requests increases. For example, the *ILP–DM*

Algorithm 1 *HEU–DM*

procedure *Input:*(G_s, G_v)
  Step 1: Find the list of candidates and their neighbors.
  for $n' \in N_v^1$ **do**
    for $n \in N_s^1$ **do**
      $d \leftarrow dis(loc(n'), loc(n))$
      if $d \leq \delta(n)$ **then**
        $candidates(n') \leftarrow n$
      end if
    end for
    for $cand \in candidates(n')$ **do**
      for $n \in N_s^1$ **do**
        $d \leftarrow dis(loc(cand), loc(n))$
        if $n \neq cand$ and $d \leq 2\delta(n)$ **then**
          $neighbor(n')(cand) \leftarrow n$
        end if
      end for
    end for
  end for
  Step 2: Find all possible combinations of candidates.
  $combs\_cands \leftarrow \emptyset$
  for $n' \in N_v^1$ **do**
    $combs\_cands \leftarrow combvec(combs\_cands, candidates(n'))$
  end for
  Step 3: Find the best combinations of candidates (minimum interference).
  $min\_intf \leftarrow \infty$
  for $comb \in combs\_cands$ **do**
    $substrate\_resources\_copy \leftarrow substrate\_resources$
    $intf\_all \leftarrow 0$
    for $n' \in N_v^1$ **do**
      $cand \leftarrow comb(n')$
      $intf(cand) \leftarrow 0$
      if $\omega^s_{ant}(cand) < \omega^v_{ant}(n')$ or $|\omega^s_{prb}(cand)| < \omega^v_{prb}(n')$ **then**
        break
      end if
      for $neigh \in neighbor(n')(cand)$ **do**
        for $prb\_idx \in \omega^s_{prb}$ **do**
          if $neigh(prb\_idx) = cand(prb\_idx) = 1$ **then**
            $intf(cand) = intf(cand) + 1$
          end if
        end for
      end for
      $intf\_all = intf\_all + intf(cand)$
      Update $substrate\_resources\_copy$
    end for
    if $intf\_all \leq min\_intf$ **then**
      $min\_intf \leftarrow intf\_all$
      $best\_cands \leftarrow comb$
    end if
  end for
  Step 4: Allocate resources & select functional splits.
  for $n' \in N_v^1$ **do**
    $mapped(n') \leftarrow best\_cands(n')$
    $sorted(best\_cands(n')) \leftarrow sortprb(intf(best\_cands(n')) \uparrow)$
    $alloc\_prb(n') \leftarrow sorted(best\_cands(n'))[1 : \omega^v_{prb}(n')]$
    for $s \in \{1, 2, ..., |R_n^{n'}|\}$ **do**
      if $intf(best\_cands(n')) \in intf\_bounds(s)$ **then**
        $split \leftarrow s$
        break
      end if
    end for
    $fh\_band(n') \leftarrow compute\_band(\omega^v_{prb}(n'), \omega^v_{ant}(n'), split)$
  end for
end procedure

algorithm takes one week on Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using the Matlab ILP solver (intlinprog) to map a request, having 20 virtual nodes, to a substrate network with 20 nodes. In order to address this scalability issue, we also propose the *HEU–DM* heuristic, which is able to embed the same virtual network request in a limited amount of time. Like in the case of the *ILP–DM* algorithm, in this case also a dynamic embedding is considered. It is worth noticing that, in order to ensure the correctness of the solutions, we pass all the solutions found by the heuristic trough the same constraints defined for the ILP formulation in Subsection V-B.

The basic design intuition behind *HEU–DM* is as follows. Initially, for each virtual DU in the virtual network request, it computes a list of candidate substrate DUs and a list of neighbor DUs for each candidate DU. Then, *HEU–DM* generates a matrix that contains all possible combinations

Algorithm 2 *HEU–ST*

procedure *Input:*(G_s, G_v)
  Step 1: Compute a list of candidates.
  for $n' \in N_v^1$ **do**
    $candidates(n') \leftarrow \emptyset$
    for $n \in N_s^1$ **do**
      $d \leftarrow dis(loc(n'), loc(n))$
      if $d \leq \delta(n)$ **then**
        if $\omega^v_{ant}(n') \leq \omega^s_{ant}(n)$ and $\omega^v_{prb}(n') \leq |\omega^s_{prb}(n)|$ **then**
          $candidates(n') \leftarrow n$
        end if
      end if
    end for
    Step 2: Select the small cell at which the inter–cell interference is minimum.
    $min\_intf \leftarrow \infty$
    for $cand \in candidates(n')$ **do**
      for $n \in N_s^1$ **do**
        $d \leftarrow dis(loc(cand), loc(n))$
        if $n \neq cand$ and $d \leq 2\delta(n)$ **then**
          $neighbor(cand) \leftarrow n$
        end if
      end for
      $intf(cand) \leftarrow 0$
      for $neigh \in neighbor(cand)$ **do**
        for $prb\_idx \in \omega^s_{prb}$ **do**
          if $neigh(prb\_idx) = cand(prb\_idx) = 1$ **then**
            $intf(cand) = intf(cand) + 1$
          end if
        end for
      end for
      if $intf(cand) \leq min\_intf$ **then**
        $min\_intf = intf(cand)$
        $best\_cand \leftarrow cand$
      end if
    end for
    Step 3: Allocate resources & select functional splits.
    $mapped(n') \leftarrow best\_cand$
    $sorted(best\_cand) \leftarrow sortprb(intf(best\_cand) \uparrow)$
    $alloc\_prb(n') \leftarrow sorted(best\_cand)[1 : \omega^v_{prb}(n')]$
    for $s \in \{1, 2, ..., |R_n^{n'}|\}$ **do**
      if $intf(best\_cand) \in intf\_bounds(s)$ **then**
        $split \leftarrow s$
        break
      end if
    end for
    $fh\_band(n') \leftarrow compute\_band(\omega^v_{prb}(n'), \omega^v_{ant}(n'), split)$
    Update substrate resources
  end for
end procedure

of candidate DUs for each virtual DU in the request (i.e., one candidate DU per virtual DU in a single combination). This is followed by considering all possible combinations and selecting the one that would introduce the lowest level of inter–cell interference in the network. Lastly, each virtual DU is mapped to its corresponding substrate DU in the selected candidate combination, the requested number of PRBs is assigned, the functional split is selected for the host DU and the fronthaul bandwidth is allocated to the virtual DU, considering its PRB requirement, RF front–end requirement and the functional split option of the host DU.

Before describing the details of how *HEU–DM* works, let us make the following notations. Let $m_1 = |N_v^1|$ and $m_2 = |N_s^1|$ be the number of, respectively, virtual and substrate DUs, while n_1 and n_2 be the maximum number of, respectively, candidate substrate DUs per virtual DU and neighbor DUs per candidate DU. Finally, let $s = |R_n^{n'}|$ be the number of functional split options, and $p = |\omega^s_{prb}|$ be the number of PRBs at each substrate small cell.

HEU–DM is composed of four steps. In the first step, a cluster of candidate DUs $candidates(n')$ is selected for each virtual DU $n' \in N_v^1$ by considering its desired location. Then, for each candidate substrate DU $cand \in candidates(n')$ of each virtual DU $n' \in N_v^1$, a neighbor list is created $neighbor(n')(cand)$. This list contains all the substrate DUs that have an overlapping coverage area with the candidate DU $cand \in candidates(n')$. This process takes $O(m_1 m_2(n_1+1))$ time. In the second step, by using $combvec()$ function in Matlab®, a combination matrix $combs\_cands$ is created, covering the entire search space of the candidate combinations for all the virtual DUs. Each column of this matrix represents one combination of candidate substrate DUs i.e. one candidate DU per virtual DU. This process requires $O(m_1)$ time.

The third step aims at finding the best combination of candidates (i.e., the best candidate column vector); that is, the one that after mapping the virtual DUs upon would introduce the lowest level of inter–cell interference in the network. Specifically, each candidate vector $comb$ is considered, and for each candidate substrate DU $cand$ of each virtual DU $n' \in N_v^1$ in the candidate vector $comb$, RF front–end and PRB resource availability is checked in order to find out whether the considered substrate DU is capable of supporting the resource requirements of the virtual DU. The total inter–cell interference is then estimated by accumulating the inter–cell interference at each candidate substrate DU in the $comb$ vector, which is computed by checking the usage of each pair of PRB (i.e., on the candidate $cand$ and its each neighbor $neigh$) in the PRB set ω^s_{prb}. At the end of this step, the combination vector $best\_cands$ is picked that would introduce the lowest level of network–wide inter–cell interference. Step 3 requires $O(m_1^2 n_1 n_2 p)$ time.

In the last step, each virtual DU $n' \in N_v^1$ is mapped to its corresponding candidate substrate DU in the combination vector $best\_cands(n')$. The PRBs of the host DUs are sorted in the ascending order of likelihood in terms of interference, and then the requested number of PRBs $\omega^v_{prb}(n')$ is allocated starting from the one that would introduce the lowest level of inter–cell interference in its collision domain. After the PRBs have been allocated, the overall inter–cell interference level at the host DU is estimated, and the appropriate functional spit is selected. This is followed by computing the fronthaul bandwidth required to host the virtual DU by using the $compute\_band()$ function, providing inputs the requested number of PRBs, the RF front–ends and the split option of the host substrate DU. The last step takes $O(m_1 s)$ time. Thus, the overall time complexity of the *HEU–DM* heuristic is $O(m_1[m_2(n_1+1) + m_1 n_1 n_2 p + s + 1])$.

E. Scalable static placement heuristic (HEU–ST)

The scalability might become a problem also for the *HEU–DM* heuristic when big–sized substrate/virtual networks with a few hundreds of substrate/virtual nodes are considered. In order to address this problem, we also propose a real–time *HEU–ST* heuristic. As the name of the heuristic implies, *HEU–ST* embeds statically the virtual network requests. In other words, with the arrival of a new virtual network request, only that request is embedded.

The basic design intuition behind *HEU–ST* is as follows. Initially, *HEU–ST* considers the first virtual DU in the virtual network request, creating a list of its candidate DUs. It then loops over candidate DUs and for each creates a list of neighbor DUs, which is then considered in order to find the candidate DU that after mapping the virtual DU upon would introduce the lowest level of inter–cell interference in its collision domain. Lastly, the considered virtual DU is mapped to the selected candidate substrate DU, the requested number of PRBs is assigned, the functional split is selected for the host DU and the fronthaul bandwidth is allocated to the virtual DU, considering its PRB requirement, RF front–end requirement, and the functional split option of the host substrate DU. The substrate resources are then updated and the described process is repeated for the rest of the virtual DUs in the request.

We will now describe the details of how *HEU–ST* works, using the same notations for both heuristics in order to estimate the embedding time complexity. *HEU–ST* is composed of three steps. In the first step, a cluster of candidate DUs is selected for each $n' \in N_v^1$ virtual DU, considering its requirements in terms of the antenna configuration, the number of PRBs and the desired location. This step takes $O(m_1 m_2)$ time. In the second step, each candidate DU $cand \in candidates(n')$ of each virtual DU $n' \in N_v^1$ is considered, and a neighbor list $neighbor(cand)$ is created for each candidate DU. The relative distance between the potential interfering DUs is considered for populating the neighbor list. The heuristic then measures the interference coming from each DU in the neighbor list. At the end of this step, the best candidate DU $best\_cand$ is picked; that is, the one that would introduce the lowest level of network–wide inter–cell interference. The second step takes $O(m_1 n_1(m_2 + n_2 p))$ time.

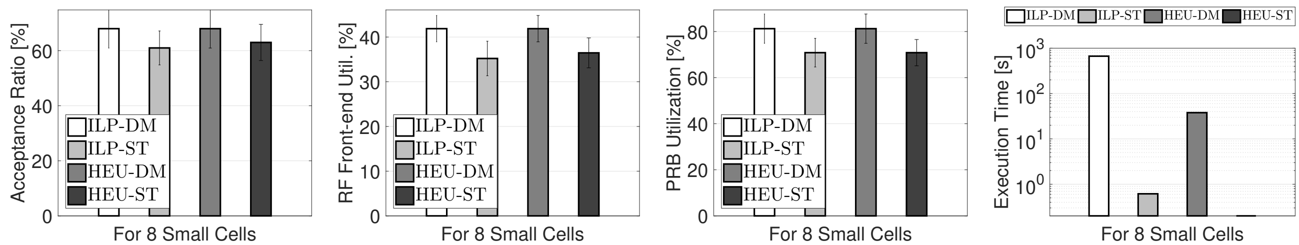
In the last step, virtual DU $n' \in N_v^1$ is mapped to the best substrate DU $best\_cand$. The PRBs of the host substrate DU are sorted in the ascending order of likelihood in terms of interference, and then the requested number of PRBs $\omega^v_{prb}(n')$ is allocated starting from the one that would introduce the lowest level of inter–cell interference in its collision domain. After the PRBs have been allocated, the overall inter–cell interference level at the host DU is estimated, and the appropriate functional split is selected. Lastly, the fronthaul bandwidth, required to host the virtual DU, is computed by using the $compute\_band()$ function, providing inputs the requested number of PRBs, the RF front–ends and the split option of the host substrate small cell. This is followed by updating the substrate resources and repeating the steps for all the virtual DUs in the virtual network request. This step takes $O(m_1 s)$ time. Thus, the overall time complexity of *HEU–ST* is $O(m_1[n_1(m_2 + n_2 p) + m_2 + s])$.

VI. Evaluation

The goal of this section is to compare the ILP–based algorithms with the heuristics. We shall first describe the simulation environment and the performance metrics used in our study. We will then report on the outcomes of the numerical simulations carried out in a discrete event simulator implemented in Matlab®.

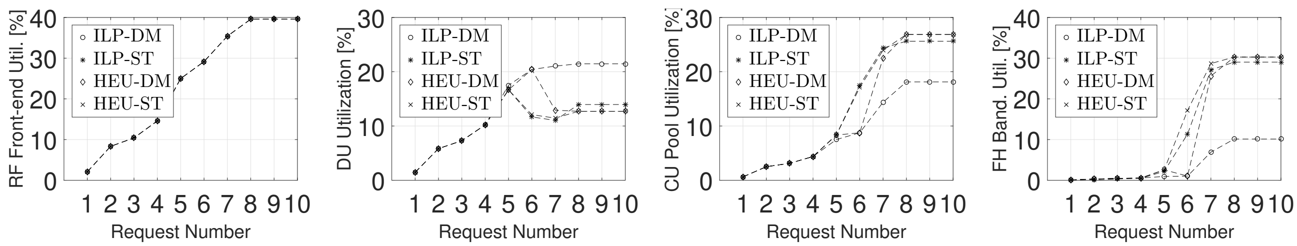
A. Simulation Environment

The reference substrate network is a star–shaped topology with 8 DUs directly connected to a single CU pool via optical fronthaul links (10 Gbps), providing mobile coverage in an area of 2 Km^2. This is a very conservative assumption, and in a more realistic scenario a ring or a tree topology may be used to connect DUs with CU pools. However, the focus of this study is on the flexible functional split, rather than on the fronthaul topology. A discussion on different fronthaul topologies is presented in Sec. VII. The inter–DU distance is 800 meters, and it is assumed that each DU possesses 6



(a) Acceptance ratio. (b) RF front–end utilization. (c) PRB utilization. (d) Execution time.

Fig. 3: Acceptance ratio, RF front–end and PRB utilizations, and the execution time for the ILP–based algorithms and the heuristics.



(a) RF front–end utilization. (b) DU utilization. (c) CU pool utilization. (d) Fronthaul bandwidth utilization.

Fig. 4: RF front–end, DU processing resource, CU pool processing resource and fronthaul (FH) bandwidth utilizations for the ILP–based algorithms and the heuristics.

TABLE V: DU and CU pool relative processing capabilities and the acceptable inter–cell interference level for the considered functional splits.

Split	$I(m)$	Processing Capacity	
		DU	CU pool
PHY–RF split	1	$0 \cdot \omega_c^s(n)$	$1 \cdot \omega_c^s(n)$
PHY split	0.6	$0.5 \cdot \omega_c^s(n)$	$0.5 \cdot \omega_c^s(n)$
MAC split	0.3	$0.7 \cdot \omega_c^s(n)$	$0.3 \cdot \omega_c^s(n)$

omni–directional[4] antennas, providing radio coverage with the radius of 500 meters. This means that in some zones there will be 200 meters of area covered by more than one DU, which in turn means that, if the users are located in that area and are connected to different DUs, being scheduled at the same PRBs, they will then create interference on one another, irrespective whether or not those users belong to the same MVNO. Lastly, it is assumed that each DU is employing LTE 20 MHz bandwidth, meaning that each DU has 100 PRBs.

One of the most prominent advantages of the PHY–RF split is that through better inter–cell coordination, it enables complex interference cancellation/avoidance algorithms such as joint transmission/reception to be employed. In some cases, however, when there is no inter–cell interference in the network, or when it is very low, employing the PHY–RF split in the C–RAN architecture would result in a waste of resources (e.g., fronthaul bandwidth) without bringing any additional benefit. For example, when DUs are well–separated, meaning that they have no overlapping coverage area, or if DUs do have an overlapping area, but the users are scheduled at different PRBs. Depending upon the level of inter–cell interference, different functional splits would be appropriate to be employed. In our model, there are three categories of interference and three corresponding functional splits (see Table V). Notice that as opposed to the PHY–RF split, in the case of the PHY/MAC split, some part of the baseband signal processing is taking place at DUs. For example, in the case of the PHY split, it is assumed that the half processing capacity is allocated to the DUs and the other half to the CU pool. This is because the most processor–hungry procedure (i.e., FFT/IFFT) is taking place in the PHY layer. The processing requirement increases at the DUs and decreases at the CU pools when a fewer layers (e.g., PHY layer, MAC layer) are centralized at the CU pools.

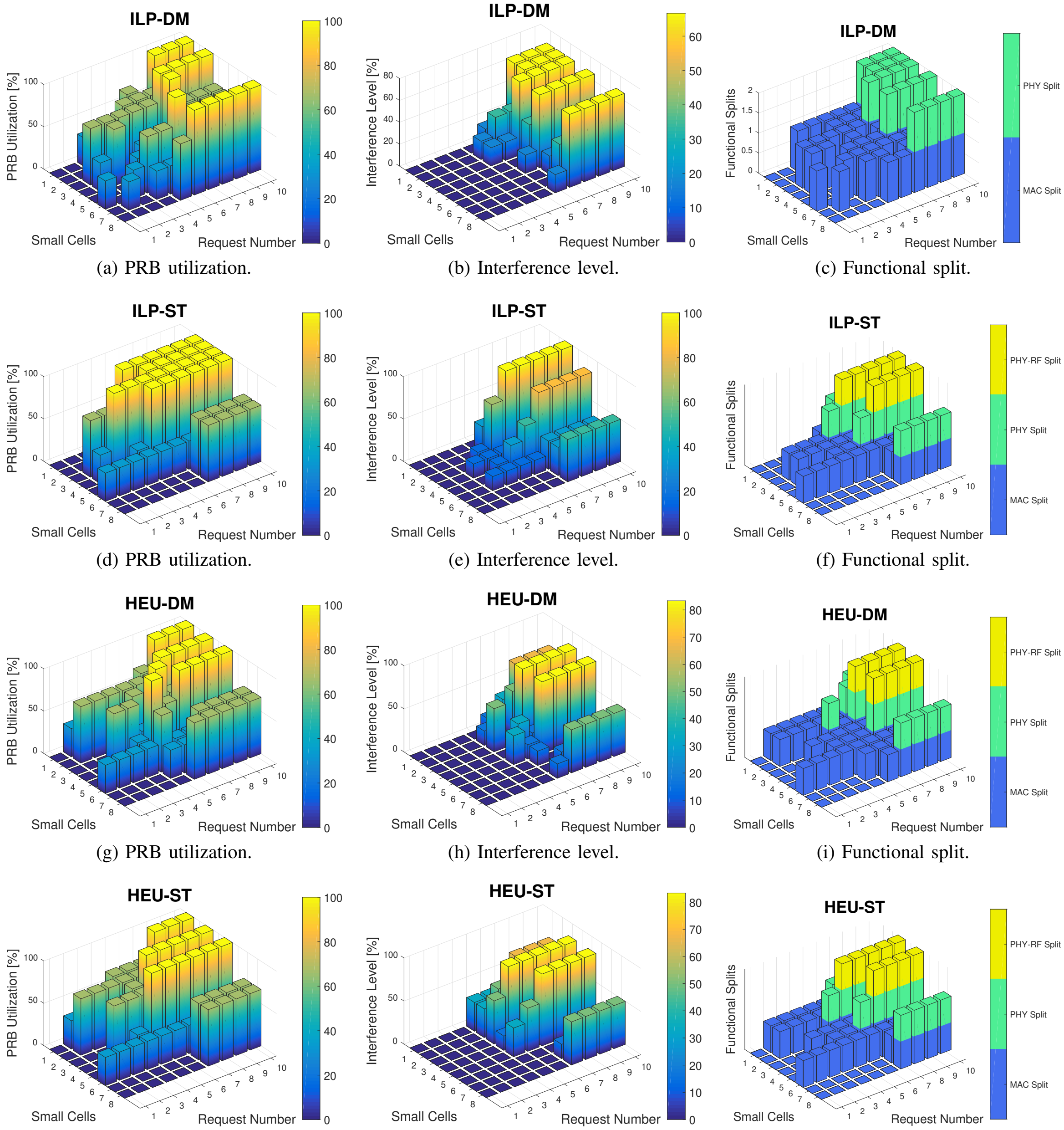
In this study, we assume that a fixed number of virtual requests are embedded sequentially. The reported results are the average of 10 simulations each with 10 embeddings with 95% confidence intervals. During each embedding process, the number of virtual DUs, RF front–ends and PRBs are randomly selected for each request in the set of, respectively, $\{1, 2\}, \{1, 2\}$ and $\{30, 60\}$.

B. Simulation Results

Figure 3 shows the acceptance ratio, the RF front–end utilization, the PRB utilization and the execution time of all the exact algorithms and the heuristics. As expected, both *ILP–DM* and *HEU–DM* dynamic embedding algorithms achieve better performance in terms of acceptance ratio, RF front–end and PRB utilization compared to their static counterparts (see Fig. 3a, 3b and 3c). We can see that both dynamic placement algorithms have accepted equal number of virtual network requests. It can be observed that also the RF front–end utilization (see Fig. 3b) and the PRB utilization (see Fig. 3c) is equal for those algorithms. These equalities prove that the *HEU–DM* heuristic achieves near optimal solutions.

With regard to the static embedding algorithms, Fig. 3a shows that the *HEU–ST* heuristic has accepted slightly more

[4]Notice that also sectoral antennas at the small cells can be easily considered. In this work we, however, we consider only omni–directional antennas since sectoral antennas, although more prevalent, would complicate the model without bringing any significant benefit.



(a) PRB utilization. (b) Interference level. (c) Functional split.

(d) PRB utilization. (e) Interference level. (f) Functional split.

(g) PRB utilization. (h) Interference level. (i) Functional split.

(j) PRB utilization. (k) Interference level. (l) Functional split.

Fig. 5: PRB utilization, interference level and functional splits of the ILP–based algorithms and of the heuristics.

virtual network requests than the *ILP–ST* algorithm. This negligible difference can be seen also in terms of the RF front–end utilization. Whereas, it can be observed that their PRB utilization is equal. This again witnesses about the *HEU–ST* heuristic achieving a good approximation compared to its *ILP–ST* counterpart. However, as we will see in Fig. 4, this does not mean that the *HEU–ST* heuristic finds more optimal solutions in comparison with the *ILP–ST* algorithm. The rationale behind *HEU–ST* performing slightly better in terms of the acceptance ratio and the RF front–end utilization is that, since for those algorithms a static embedding is taking place, it cannot be claimed that the performance of the *ILP–ST* algorithm after all embeddings must *always* be better compared to the *HEU–ST* heuristic because they have different logic to embed the virtual network requests. Moreover, by performing a static embedding, with the arrival of a new virtual network request, the previously embedded request cannot be re–embedded. Thus, the performance of the static embedding algorithm/heuristic merely depends on their previous mappings and on the requirement of the new requests in terms of the desired location, number of PRBs and RF front–ends.

The *ILP–DM* and *ILP–ST* exact algorithms become computationally intractable when networks with, respectively, a few tens and a few hundreds of substrate/virtual nodes are considered. Depending on the size of the substrate and virtual networks (e.g., ultra–dense networks with hundreds of small cells), also the *HEU–DM* heuristic might encounter a scalability problem in terms of the required time to map virtual network requests. With the sole purpose of tackling this problem, a static embedding heuristic (*HEU–ST*) has also been proposed. Figure 3d displays the total time taken to embed 10 virtual network requests (a single iteration). Note that during that iteration all the algorithms/heuristics have embedded equal number of request (see Fig. 4). It can be observed that among the dynamic embedding algorithm/heuristic, *HEU–DM* has taken 20 times less time to embed those requests compared to its exact counterpart. Whereas among the static embedding algorithm/heuristic, *HEU–ST* has embedded the virtual network requests twice faster than the *ILP–ST* algorithm. The embedding time for the ILP-based algorithms will increase exponentially with a larger substrate and virtual networks.

In order to get a better insight into how the resources of the substrate network are exploited, we will now examine the same single iteration (i.e., 10 embeddings). Figure 4 depicts the RF front–end utilization, the processing resource utilization of the DUs and the CU pool, and the overall fronthaul bandwidth utilization for *ILP–DM*, *ILP–SM*, *HEU–DM* and *HEU–ST*. In Fig. 4a, it can be seen that all algorithms/heuristics have embedded equal number of virtual network requests, and therefore, have equally utilized the RF front–end resources. In particular, all algorithms/heuristics have successfully embedded the virtual network requests up to the 8^{th} embedding, while, rejected the last two virtual network requests.

With regard to the DU processing resource utilization (see Fig. 4b), we can observe that the *ILP–DM* algorithm keeps increasing the processing resource utilization gradually, and therefore, exhibits the best performance among all the algorithms/heuristics. In essence, with the arrival of virtual network requests, this increase means that, due to the optimal embedding of all the virtual network requests, only MAC and PHY layer splits are being employed by the substrate small cells/DUs, since among the considered splits, only those splits require processing resource at the DUs. It can also be observed that the second best performance achieves the *HEU–DM* heuristic, since as opposed to the static embedding algorithm/heuristic, which reduce the DU processing resource utilization from the 6^{th} embedding, it reduces the DU processing resource utilization from the 7^{th} embedding. This essentially means that in the case of the *HEU–DM* heuristic, most of the substrate nodes employ either MAC or PHY layer splits up to the 6^{th} embedding. While from the 7^{th} embedding some of the substrate nodes start changing the splits from the PHY/MAC split to the PHY–RF split in order to be able to exploit advanced algorithms aiming at reducing/canceling the inter–cell interference.

Regarding to the static embedding algorithm/heuristic, it can be seen that their performances resemble each other in terms of the DU processing resource utilization. However, we can see that the *ILP–ST* algorithm ultimately achieves slightly higher processing resource utilization at the DUs. This proves that the *ILP–ST* algorithm has found optimal embedding solutions, and therefore, has lead to more substrate nodes employing higher–layer functional splits. In essence, this means that the inter–cell interference level at those nodes is lower from the inter–cell interference threshold starting from which the PHY–RF split should be employed.

The picture is totally different for all the algorithms/heuristics in terms of the processing resource utilization at the CU pool (see Fig. 4c). The first observation is that for all the algorithms/heuristics the processing resource utilization at the CU pool increases, regardless of the split options employed by the substrate DUs. The rationale behind this is that depending on the inter–cell interference level at the substrate small cells, which increases with the arrival of new virtual network requests, the functional splits at the small cells change from the higher–layer splits toward the lower–layer splits and, with this change, the processing resource utilization increases at the CU pool and decreases at the DUs. Since in the case of the *ILP–DM* algorithm, only the MAC split or the PHY split is employed by the substrate small cells (we will see this in Fig. 5c), the processing resource utilization at the CU pool is the lowest compared to the rest of the algorithm/heuristics. This is because, for those splits, the processing resource requirement at the CU pool is lower compared to the processing resource requirement at the CU pool for the PHY–RF split in which all baseband signal processing is taking place only at the CU pool. In the cases of *ILP–ST*, *HEU–DM* and *HEU–ST*, it can be observed that they achieve higher processing resource utilization at the CU pool as opposed to ones at the DUs. Moreover, *ILP–ST* and *HEU–ST* increase the processing resource utilization at the CU pool before the *HEU–DM* heuristic. Thus, the more are the substrate DUs that employ lower–layer the PHY split or the PHY–RF split, the more is the processing resource utilization at the CU pool and the less is the processing resource utilization at the DUs.

The fronthaul bandwidth utilization for all the algorithms/heuristics (see Fig. 4d) somewhat resembles the processing resource utilization at the CU pool. This is because apart from the processing resource utilization at the CU pool, also the fronthaul bandwidth utilization increases with the increase in the number of substrate nodes that employ the PHY split or the PHY–RF split. We can observe that the fronthual bandwidth utilization for *ILP–DM* and *HEU–DM* is very low until the 6^{th} embedding with a small spike at the 5^{th} embedding for the *HEU–DM* heuristic. The rationale behind this is that at the 5^{th} embedding one of the substrate nodes starts using the PHY split, while at the 6^{th} embedding, the number of the host substrate nodes increases, and they all use the MAC split (this can be seen in Fig. 5i). We can also observe that for the static embedding algorithm/heuristic the fronthaul bandwidth utilization is very low up to the 4^{th} embedding since at this point all the host substrate nodes are using the MAC split, which has very low fronthaul bandwidth requirement; while from the 5^{th} embedding, the fronthaul bandwidth utilization starts increasing exponentially. This huge difference in the fronthaul bandwidth utilization is due to the significant difference in the fronthaul bandwidth requirements of the splits (see Table I).

We will now examine the PRB utilization, the inter–cell interference level and the functional split at all the substrate small cells for all the algorithms/heuristics for a single iteration in order to better understand their relationship (see Fig. 5). It can be observed that the PRB utilization of individual substrate small cells for *ILP–DM*, *ILP–ST*, *HEU–DM* and *HEU–ST* varies due to different mapping decisions (see Fig. 5a, 5d, 5g and 5j). However, the network–wide PRB utilization (the sum of the PRB utilization of the small cells) is the same for all the algorithms/heuristics during each embedding. This is justified by the fact that all the algorithms/heuristics have identically accepted or rejected the virtual network request (see Fig. 4a).

Figures 5b, 5e, 5h and 5k display the inter–cell interference level in the cases of employing, respectively, *ILP–DM*, *ILP–ST*, *HEU–DM* and *HEU–ST*. It can be observed that both dynamic embedding algorithm/heuristic successfully embed up to the 4^{th} virtual network request without creating inter–cell interference in the network. Whereas, from the 5^{th} embedding the *ILP–DM* algorithm creates little amount of inter–cell interference at two substrate nodes, while the *HEU–DM* heuristic creates inter–cell interference at three substrate nodes. Moreover, it can be seen that at two of them the level of inter–cell interference is much higher compared to the case of employing the *ILP–DM* algorithm. After all the embeddings, we can see that the total inter–cell interference in the network in the case of employing the *ILP–DM* algorithm is much lower than the one in the case of employing the *HEU–DM* heuristic (notice the difference in the scales). This again proves that the mapping efficiency of the *ILP–DM* algorithm is higher compared to the one of *HEU–DM*.

The picture is different in the cases employing static embedding algorithm/heuristic. Initially, the *HEU–ST* heuristic is as efficient as both dynamic embedding algorithm/heuristic, since like them, *HEU–ST* successfully embeds up to the 4^{th} virtual network request without creating inter–cell interference in the network. Nevertheless, after it starts introducing inter–cell interference at some of the substrate small cells. Whereas, the *ILP–ST* algorithm starts creating inter–cell interference from the 3^{th} embedding, and ultimately, results in a higher network–wide inter–cell interference compared to the *HEU–ST* heuristic. As it has been already mentioned, in this case, this is a consequence of the *HEU–ST* heuristic being slightly more efficient than the *ILP–ST* algorithm.

Finally, let us analyze how the functional splits change at the substrate small cells as a function of changing inter–cell interference. Figures 5c, 5f, 5i and 5l show the functional splits per substrate small cell for a single iteration (10 embeddings) for, respectively, *ILP–DM*, *ILP–ST*, *HEU–DM* and *HEU–ST*. In general, the lower is the inter–cell interference level, the higher–layer is the selected split, leading to a more efficient fronthaul bandwidth utilization. Among all the algorithms/heuristics, the superiority of the *ILP–DM* algorithm is obvious since, thanks to fact that it is always able to find the optimal mapping for all the virtual network requests, the substrate small cells only employ either the MAC split or the PHY split. Thus, being able to keep the level of inter–cell interference low from the threshold starting from which the PHY–RF split must be employed, no substrate small cell employs the PHY–RF split. It is also obvious that the *HEU–DM* heuristic exhibits the second best performance. Whereas, the performance of *ILP–ST* resembles the performance of *HEU–ST*. The reason for this is twofold. First, if a substrate small cell has been used for mapping then by default, in the considered scenario, it employs the MAC split even if there is no inter–cell interference. We can see that up to the 4^{th} embedding there is no inter–cell interference in the network in the case of employing the *HEU–ST* heuristic (see Fig. 5h). However, as it can be seen in Fig. 5i, the substrate nodes, which have been used to map the virtual network requests up to the 4^{th} embedding, employ the MAC split. Second, although the level of inter–cell interference at some substrate nodes are higher in the case of *ILP–ST* algorithm compared to the one of *HEU–ST* heuristic, they use the same functional split option. This is because for each of the splits, there is an inter–cell interference range[5] defined (see Table V), and

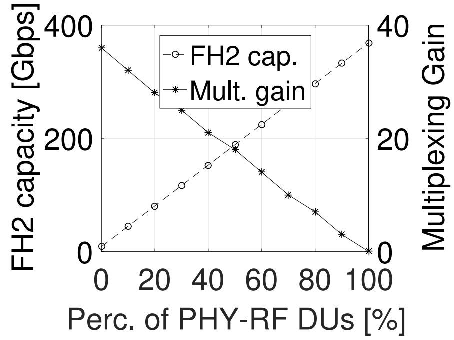


Fig. 6: FH2 link capacity requirement and the multiplexing gain as a function of DUs that employ different functional splits in the tree substrate topology.

if the level of inter–cell interference at those nodes is within the same range then regardless of the difference in the level of inter–cell interference, the same corresponding functional split is to be employed by those substrate nodes.

C. Discussion

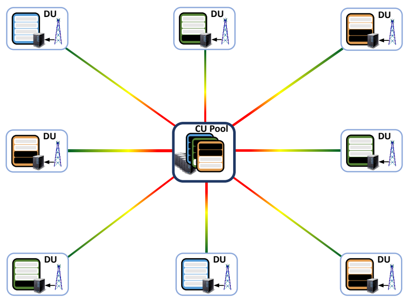
The proposed algorithms/heuristics provide MNOs with various options to select between promptness, mapping optimality and scalability. We have seen that the *ILP–DM* algorithm, thanks to its dynamic embedding strategy, achieves the optimal mapping for all the virtual network requests. However, it comes at the expense of significantly long embedding time (see Fig. 3d), which makes this algorithm not applicable to dense mobile networks with a few tens of small cells in the substrate/virtual networks. On the other hand, the *HEU–DM* heuristic, although less efficient, approximates the optimal mapping solutions found by the *ILP–DM* algorithm. Moreover, it is significantly faster in embedding virtual network requests, which makes it more scalable compared to *ILP–DM*. *HEU–DM*, however, is not applicable to ultra–dense mobile networks with a few hundreds of small cells in the substrate/virtual networks. If it is decided to employ a dynamic embedding algorithm/heuristic, one must also take into account the possible downsides (e.g., service interruption of the users of MVNOs) of the re–embedding of the virtual network requests.

In order to address the scalability problems of the exact dynamic placement algorithm and the dynamic placement heuristic, an exact static embedding algorithm (*ILP–ST*) and a static embedding heuristic (*HEU–ST*) are also proposed. We have seen that the static embedding algorithm/heuristic is less efficient in embedding virtual network requests and is, therefore, less efficient in employing the network resources compared to their dynamic counterparts. We have also seen that *ILP–ST* and *HEU–ST* are not comparable since their embedding decisions depend on their previous embedding, and they use different strategies to embed the virtual network requests. Thus, one must consider the availability of the network resources and the embedding time requirement in order to select the embedding algorithm/heuristic that would be the most suitable for their network.

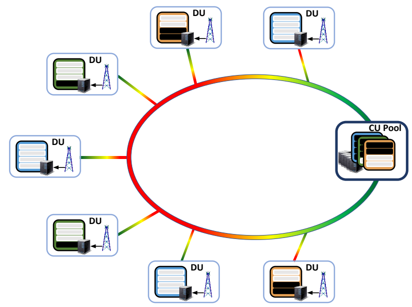
VII. PHY–RF Split Vs. Flexible Split

In this section, we will compare the PHY–RF split with the flexible functional split for the physical network topologies

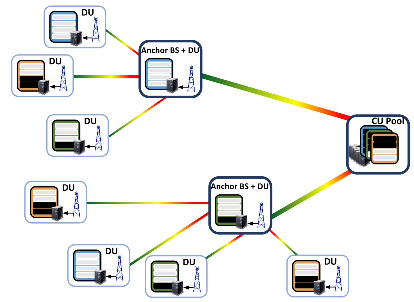
[5]The inter–cell interference value $I(m)$ for each functional split defined in Table 4 puts un upper bound to the acceptable inter–cell interference range since those values are used in the inequality constraint (12) in the ILP formulation.



(a) Star topology.



(b) Ring topology.



(c) Tree topology.

Fig. 7: Star, ring and tree substrate network topologies.

depicted in Fig. 7. Since the pros and cons of the PHY–RF split is well investigated in the literature, we will highlight the pros and cons of employing the flexible functional split compared to the PHY–RF split.

A. Pros of the flexible functional split over the PHY–RF split

The pros of flexibly selecting the functional splits compared to the PHY–RF split are manifold. However, due to space limitation, we will discuss most prominent advantage; that is, the efficient usage of the fronthaul bandwidth resources. In the case of the star substrate topology, in which the DUs are connected to the CU pool by means of direct optical links (see Fig. 7a), there is no benefit on the fronthaul links since each of the links has to have enough capacity to support the PHY–RF split. However, in the cases of a ring or a tree topology, in which the DUs are connected to the CU pool, respectively, through a high–capacity fronthaul ring, and through anchor Base Stations (BSs) and a high–capacity fronthaul links, the advantage of using flexible functional splits is significant. By exploiting the multiplexing gain of different functional splits, the capacity of the fronthaul links (i.e., the ring link in Fig. 7b and the links from the anchor BSs to the CU pool in Fig. 7c) can be estimated. The less is the difference in the number of DUs that use different splits, the more efficient is the link utilization. This will result in significant savings in terms of CAPEX and OPEX while deploying the optical fronthaul network, which incurs the highest portion of the total investment required to deploy the C–RAN architecture.

In order to illustrate this, let us first make the following assumptions. We consider flexible functional splits in a tree[6] substrate network topology like the one depicted in Fig. 7c, which is composed of 50 DUs that are connected to a single CU pool through an anchor BS. The FH network for each DU is composed of two parts: the link from the DU to the anchor BS (called $FH1$) and the link from the anchor BS to the CU pool (called $FH2$). The latter is of interest to us since we want to analyze the multiplexing gain of the considered splits on the link from the aggregation point (i.e. the anchor BS) to the CU pool. It is assumed that optical FH links are used and that the $FH1$ links have enough capacity to support the bandwidth requirement of the PHY–RF split; while the single $FH2$ link has enough capacity to support the FH bandwidth requirements of the DUs in the case they all only employ the PHY–RF split. It is also assumed that Wavelength Division Multiplexing Passive Optical Network (WDM–PON) technology is used and that each of the lightpaths/wavelengths supports 10Gbps traffic. Thus, since we want to study the impact of the flexible functional split on the FH deployment cost, this cost for us is just the cost of setting up lightpaths based on the FH bandwidth requirement of the considered functional splits. Lastly, it is assumed that each DU is composed of 3 sectors each supporting 2×2 MIMO configuration employing LTE 20MHz bandwidth, which is fully utilized by the UEs.

Under these assumptions, let us analyze Fig. 6, which shows the $FH2$ link capacity requirement and the multiplexing gain for DUs that may use any of the considered functional splits at the given time in the case of flexibly employing the considered functional splits. The multiplexing gain is expressed in terms of the required number of lightpaths in order to support the $FH2$ link capacity requirement. This can be easily translated to CAPEX by multiplying the multiplexing gain value by the cost of creating a lightpath. The x–axis shows the percentage of the DUs that are employing the PHY–RF split, while the rest of the DUs are flexibly using either the PHY split or the MAC split. The trade–off between the $FH2$ link capacity and the multiplexing gain is obvious. The more is the number of DUs employing the PHY–RF split at the given time, the more is the $FH2$ link capacity requirement but also the less is the multiplexing gain, and therefore, so is the CAPEX savings. One needs to find the *right* proportions of DUs that use different functional splits (e.g., the PHY–RF split, the PHY split or the MAC split), the traffic aggregation of which would provide the highest multiplexing gain without violating the capacity limit of $FH2$ link. However, notice that among all the DUs, only a few of them employ the PHY–RF split, even though it may provide a huge multiplexing gain, it might result in some of the lightpaths being underutilized. Whereas, having too many DUs that employ the PHY–RF split, although increases the utilization of the $FH2$ link, might result in a less multiplexing gain, and therefore, less CAPEX savings.

B. Cons of the flexible functional split over the PHY–RF split

The mentioned advantage of flexibly employing functional splits comes at the expanse of the network design complexity. In the case of only employing the PHY–RF split, the design of DUs is very simple and cheap. Being compact units, DUs can easily be deployed, for example, on street furnitures such as on lamp posts or on billboards. The flexible functional split offsets this advantage since it has to be able to flexible support

[6]The result is the same also for the ring topology where the fronthaul bandwidth estimation would be for the ring link.

in our case the PHY–RF split, the PHY split and the MAC split. Thus, both DUs and CU pools have to support all the functionalities of the considered splits. This would require site rental cost, and therefore, increase the CAPEX.

Another cons of flexible functional splits is that it requires a new transport protocol design that can flexibly support different splits. A universal frame format and data plane need to be designed, which can construct frames and transmits over the FH network, regardless of the selected functional split option. However, we believe that the flexibility of a functional split option selection and the fronthaul network cost savings will payoff the challenges posed by the flexibility in selecting the functional split.

VIII. Conclusions

Flexible functional split in the 5G RAN provides the possibility of exploiting complex CoMP algorithms designed to reduce/cancel the inter–cell interference. However, depending upon the actual level of interference, different functional splits may be used. We have seen that the processing requirements of the DUs and the CU pool, and fronthaul bandwidth requirement change substantially, depending upon the selected functional split option. This means that significant benefits can be reaped by employing the *right* functional split option for each small cell. Although, in our scenario the functional splits change from the higher–layer splits (e.g., the MAC split) toward the lower–layer splits (e.g., the PHY split, the PHY–RF split), the functional splits can also be changed towards the reverse direction, for example, considering daylight vs. night traffic variation and users distribution.

As a future work, we plan to extend the problem formulation to real scenarios. In particular, we want to consider an operational LTE–A mobile network in which both wireless and optical links are used as transport mediums. Based on the availability of the transmission links as well as the spatially and temporarily fluctuating traffic demand at eNBs, we want to study flexible functional split options that can be applied to different parts of mobile networks in different parts of a day.

References

[1] NOKIA, "Ultra Dense Network (UDN)," 2016. [Online]. Available: http://tools.ext.nokia.com/asset/200295

[2] K. Chen and R. Duan, "C-RAN the road towards green RAN," *China Mobile Research Institute, white paper*, vol. 2, 2011.

[3] "Technical Specification Group Radio Access Network; Study on New Radio Access Technology; Radio Access Architecture and Interfaces ," 3GPP TR 38.801 V2.0.0, Tech. Rep., 2017.

[4] S. C. Virtualization, "Functional splits and use cases," in *Small Cell Forum release*, vol. 6, United Kingdom, 2016.

[5] J. Huang and Y. Yuan, "White paper of next generation fronthaul interface," *labs. chinamobile. com/cran, ver*, vol. 1, 2015.

[6] "Further study on critical C-RAN technologies," NGMN, Tech. Rep., 2015.

[7] "Common Public Radio Interface, Interface Specification V6.0," CPRI, Tech. Rep., August 2013.

[8] D. Harutyunyan and R. Riggio, "Flexible Functional Split in 5G Networks," in *Proc. of IEEE CNSM*, Tokyo, Japan, 2017.

[9] S. Namba, T. Matsunaka, T. Warabino, S. Kaneko, and Y. Kishi, "Colony-RAN architecture for future cellular network," in *Proc. of IEEE FutureNetw*, Berlin, Germany, 2012.

[10] N. Carapellese, M. Tornatore, and A. Pattavina, "Placement of base-band units (BBUs) over fixed/mobile converged multi-stage WDM-PONs," in *Proc. of IEEE ONDM*, Brest, France, 2013.

[11] N. Carapellese, M. Tornatore, and A. Pattavina, "Energy-efficient baseband unit placement in a fixed/mobile converged WDM aggregation network," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 8, pp. 1542–1551, 2014.

[12] H. Holm, A. Checko, R. Al-obaidi, and H. Christiansen, "Optimal assignment of cells in C-RAN deployments with multiple BBU pools," in *Proc. of EuCNC*, Paris, France, 2015.

[13] R. Al-obaidi, A. Checko, H. Holm, and H. Christiansen, "Optimizing Cloud-RAN deployments in real-life scenarios using Microwave Radio," in *Proc. of EuCNC*, Paris, France, 2015.

[14] C. Aleksandra, H. L. Christiansen, and M. S. Berger, "Evaluation of energy and cost savings in mobile Cloud RAN," in *Proc. of OPNETWORK*, 2013.

[15] M. Y. Lyazidi, N. Aitsaadi, and R. Langar, "Dynamic resource allocation for Cloud-RAN in LTE with real-time BBU/RRH assignment," in *Proc. of IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[16] B.-S. Huang, Y.-H. Chiang, and W. Liao, "Remote radio head (RRH) deployment in flexible C-RAN under limited fronthaul capacity," in *Proc. of IEEE ICC* , Paris, France, 2017.

[17] K. Samdanis and X. Costa-Perez and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[18] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed, and T. Ahmed, "Scheduling wireless virtual networks functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 240–252, 2016.

[19] X. Zhou and R. Li and T. Chen and H. Zhang, "Network slicing as a service: enabling enterprises' own software-defined cellular networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.

[20] M. Richart and J. Baliosian and J. Serrat and J. L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[21] I. da Silva and G. Mildh and A. Kaloxylos and P. Spapis and E. Buracchini and A. Trogolo and G. Zimmermann and N. Bayer, "Impact of network slicing on 5G Radio Access Networks," in *Proc. of EuCNC*, Athens, Greece, 2016.

[22] P. Rost, C. J. Bernardos, A. D. Domenico, M. D. Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wbben, "Cloud technologies for flexible 5G radio access networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 68–76, May 2014.

[23] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN," *Signal Processing Magazine, IEEE*, vol. 31, no. 6, pp. 35–44, 2014.

[24] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 105–128, 2013.

[25] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *Wireless Communications, IEEE*, vol. 22, no. 5, pp. 105–111, 2015.

[26] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based framework for flexible baseband function splitting and placement in C-RAN," in *Proc. of IEEE ICC*, London, United Kingdom, 2015.

[27] A. Maeder, M. Lalam, A. De Domenico, E. Pateromichelakis, D. Wubben, J. Bartelt, R. Fritzsche, and P. Rost, "Towards a flexible functional split for Cloud-RAN networks," in *Proc. of EuCNC*, Bologna, Italy, 2014.

[28] M. Jaber, D. Owens, M. A. Imran, R. Tafazolli and A. Tukmanov, "A joint backhaul and RAN perspective on the benefits of centralised RAN functions," in *Proc. of IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[29] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, "Evaluating c-ran fronthaul functional splits in terms of network level energy and cost savings," *Journal of Communications and Networks*, vol. 18, no. 2, pp. 162–172, April 2016.

[30] I. Koutsopoulos, "Optimal functional split selection and scheduling policies in 5G Radio Access Networks," in *Proc. of IEEE ICC Workshop*, Paris, France, 2017.

[31] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5G radio access network architecture based on flexible functional control/user plane splits," in *Proc. of EuCNC*, Oulu, Finland, 2017.

[32] S. Sesia, I. Toufik, and M. Baker, *LTE - the UMTS long term evolution: from theory to practice*. Wiley, 2011.

[33] Z. L. Zaki, Yasir, C. Goerg, and A. Timm-Giel, "LTE wireless virtualization and spectrum management," in *Proc. of IEEE WCNC*, Budapest, Hungary, 2010.

[34] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE mobile network virtualization," *Mobile Networks and Applications*, vol. 16, no. 4, pp. 424–432, 2011.

[35] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: a substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1333–1346, 2012.