Flexible Functional Split in 5G Networks

Davit Harutyunyan, Roberto Riggio
FBK CREATE-NET, Trento, Italy; Email: dharutyunyan,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. The VNE is formulated as an Integer Linear Programming problem whose objective is to minimize the inter–cell interference and the fronthaul bandwidth utilization by dynamically selecting the appropriate functional split. Finally, a scalable VNE heuristic is also proposed.**

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

I. Introduction

Compared to LTE and LTE–Advanced, 5G networks, are expected to deliver a 1000 times increase in the system capacity, reduced round–trip delay, and enhanced cell–edge performances. 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 Remote Radio Heads – RRH) and consolidated in large data–centers (termed Baseband Units – BBU). 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 BBUs with RRHs) where protocols like the common public radio interface (CPRI) [3] are used to carry the IQ samples over (typically) an optical fiber.

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 functional splits, each characterized by a different demarcation point between the centralized and the distributed units, 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 efficiently employ the fronthaul bandwidth and baseband processing resources.

In this paper, we formalize and solve a dynamic 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 dynamically embedded by the infrastructure providers (InPs). The objective is to select the functional split option that jointly minimizes inter–cells interference *and* fronthaul bandwidth utilization. We also propose a heuristic in order to tackle the scalability problem of the ILP algorithm. Lastly, we compare the ILP and heuristic using a numerical simulator.

The rest of this paper is structured as follows. The related work is discussed in Sec. II. The different functional splits are introduced in Sec. III. The substrate network and the virtual network request models are detailed in Sec. IV. The ILP problem is formulated in Sec. V. The numerical results are reported in Sec. VI. Finally, Sec. VII draws the conclusions.

II. Related Work

BBU Placement. A sizeable body of work has been published on the BBU placement problem [4], [5], [6], [7], [8]. In [5], the authors propose a Colony–RAN cellular architecture

Research leading to these results received funding from the European Unions H2020 Research and Innovation Action under Grant Agreement H2020-ICT-644843 (COHERENT Project).

978-3-901882-98-2 © 2017 IFIP

which is able to change the cell layout by dynamically adapting the connections between BBUs and RRHs according to the network conditions. An optimization algorithm is presented in [6] for the BBU Placement problem over a Fixed/Mobile converged optical network. The authors formulate an ILP problem, which efficiently calculates the minimum number of required BBU pools taking into account the maximum allowed distance between RRHs and their BBUs. The same authors put forward an energy–efficient BBU Placement algorithm in optical networks in [4]. An ILP optimization problem is formalized in [7] for optimizing cells assignment to different BBU pools. Statistical multiplexing gain and required fiber length are used as key performance indicators. A two–stage mechanism is proposed in [8]. The authors formulate PRB allocation as a mixed integer linear programming (MILP) problem and use the results to formulate an RRHs–BBUs assignment problem as a Multiple Knapsack Problem (MKP), taking into account the real–time traffic load in RRHs.

Network Sharing. 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 [9] introduce an on–demand capacity broker concept, which is able to securely expose selected service features via APIs, allowing InPs to allocate required portion of their networks to MVNOs, OTTs, or vertical market players. Being inspired by the concept of "everything" as a service (XaaS) and having a goal of assisting mobile network operators to offer a customizable end–to–end service to MVNOs, Network Slice–as–a–Service (NSaaS) concept is introduced in [10]. An extensive survey on network slicing can be found in [11]. Various approaches of wireless slicing are presented for different technologies such as LTE, WiMAX and Wi–Fi. A detailed study on the impact of network slicing in 5G RANs can be found in [12].

Flexible Functional Split. The functional split problem has attracted significant attention from both the academia and the industry [13], [14], [15]. 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., traffic demand, energy efficiency, and latency constraints) have to be taken into account to decide the actual split point. A detailed discussion on various functional splits can be found in [16], [17], [18], [19], [20]. The authors of [16] 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 [17]. Based on burstiness of the traffic and the fact that the mobile traffic varies depending upon the area (e.g., residential, office) and the time of a day, mathematical and simulation methods are proposed in [21] for quantifying the multiplexing gain of C–RAN and PDCP/RLC functional splits. The authors of [22] put forward a graph–based algorithm for analyzing the different baseband functional splits. The authors of [18] explore all the possible wired/wireless transport fronthauling technologies as well as the associated bandwidth and latency requirements for the different functional splits. A detailed investigation on the various functional splits at the PHY layer is conducted in [19], [20]. A case–study analysis is presented in [23] 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, C–RAN with optical fiber fronthaul option is the most profitable one, though it incurs the highest deployment costs. Recently, significant effort has been made towards packet–based architecture in fronthaul networks [13]. Different PHY–layer functional splits are studied in [24] for packet–based fronthaul networks.

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

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

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 different functional splits in terms of fronthaul bandwidth and latency requirements [25].

C–RAN. The traditional C–RAN split corresponds to full resource centralization with all baseband signal processing taking place at the BBU pool, leaving the RF functions (e.g., analogue–to–digital and reverse conversion, signal amplification) at the RRH side. While this functional split provides several advantages in terms of energy efficiency, computational diversity, 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 RRHs, it is possible to significantly relax the fronthaul requirements in terms of both bandwidth and latency. As it can be seen from Table I, taking the requirements of C–RAN as a baseline, the PHY split allows the fronthaul bandwidth and latency requirements to be reduced by a factor of 2.5 and 8 respectively. Notice how, fronthaul requirements are relaxed at the expense of the resource centralization gain. For example, compared to C–RAN, CoMP can no longer be employed with a PHY split.

MAC Split. In this case the HARQ procedure is taking place at the RRH while the rest of the MAC functions along

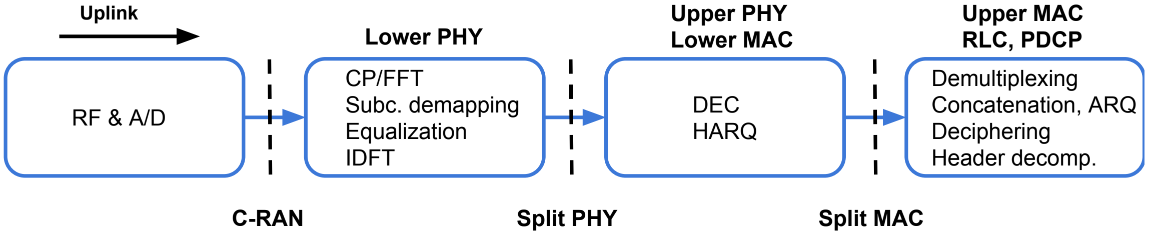


Fig. 1: Signal processing along with some of the functional split options within the RAN protocol stack in LTE networks.

with the upper layers are consolidated at the BBU pool. Compared to the C–RAN 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

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. On the lower left part of the figure we can see the traditional D–RAN architecture in which the RRH and the BBU are deployed in close proximity. As opposed to the D–RAN case, for all the other functional splits, the RRHs are decoupled from the BBU pool, and a wired/wireless fronthaul is used for their interconnection. Notice that, in the cases of the PHY and MAC splits, also the RRHs are equipped with processing capabilities.

A. Substrate Network Model

Let $G_s = (N_s, E_s)$ be an *undirected* graph modelling the physical network, where $N_s = N_s^1 \cup N_s^2$ is the set of $n_1 = |N_s^1|$ RRH sites and $n_2 = |N_s^2|$ BBU 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 number of physical resource blocks (PRBs) 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 centred on RRH n. 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.

TABLE II: Substrate network parameters

Variable	Description
G_s	Substrate network graph.
N_s	Substrate nodes in G_s.
N_s^1	Substrate RRH sites in G_s.
N_s^2	Substrate BBU pools in G_s.
E_s	Substrate links in G_s.
$I(m)$	Interference level range for m^{th} functional split.
$\omega^s_{ant}(n)$	Number of RF front–ends available at RRH $n \in N_s^1$.
$\omega^s_{prb}(n)$	Number of PRBs available at RRH site $n \in N_s^1$.
$\omega^s_{prc}(n)$	The processing capacities of the nodes $n \in N_s$.
$\omega^s_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_s$ (in Gbps).
$loc(n)$	Geographical location of node $n \in N_s$.
$\delta(n)$	Coverage radius of node $n \in N_s$ (in meters).
R^n	The set of fronthaul bandwidths of node $n \in N_s$.

Notice how it is our assumption that the BBU pool is equipped with enough computational capacity to support all RRHs employing the highest possible functional split i.e., C–RAN, which requires all baseband signal processing to take place at the BBU pool. Whereas, the RRHs are equipped with enough computational capacity to process the signals with the lowest possible functional split, i.e. the MAC split. The RRHs are also equipped with RF front–ends, as opposed to the BBU pool, and are connected to the BBU pool by means of wired/wireless fronthaul links.

B. Virtual Network Request Model

There are different approaches to model virtual network requests, from resource–based models [26] [27] to throughput–based models [28]. In this work, a resource–based model is used. According to this model, 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. Notice how, 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 modelled as *undirected* graphs $G_v = (N_v, E_v)$ where $N_v = N_v^1 \cup N_v^2$ is the set of $n_1 = |N_v^1|$ virtual RRH sites and $n_2 = |N_v^2|$ virtual BBU pools and $E_v \subseteq N_v^1 \times N_v^2$ is the set of virtual fronthaul links. Notice that MVNOs do not request BBU pool capacity nor they specify the small cell fronthaul requirements. Nodes in the virtual network request have two weights $\omega^v_{ant}(n)$ and $\omega^v_{prb}(n)$ indicating, respectively, the number of RF front–ends and the number of PRBs requested for each node $n \in N_v^1$. Given the chosen functional split and considering the aforementioned parameters $(\omega^v_{ant}(n), \omega^v_{prb}(n))$, the fronthaul bandwidth required to support a given small cell as well as the BBU Pool capacity can easily be derived [25]. Each RRH site $n \in N_v^1$ is also associated with a geographic location $loc(n)$, as x, y coordinates. This information together with the substrate node location and its coverage radius is used to express how far a

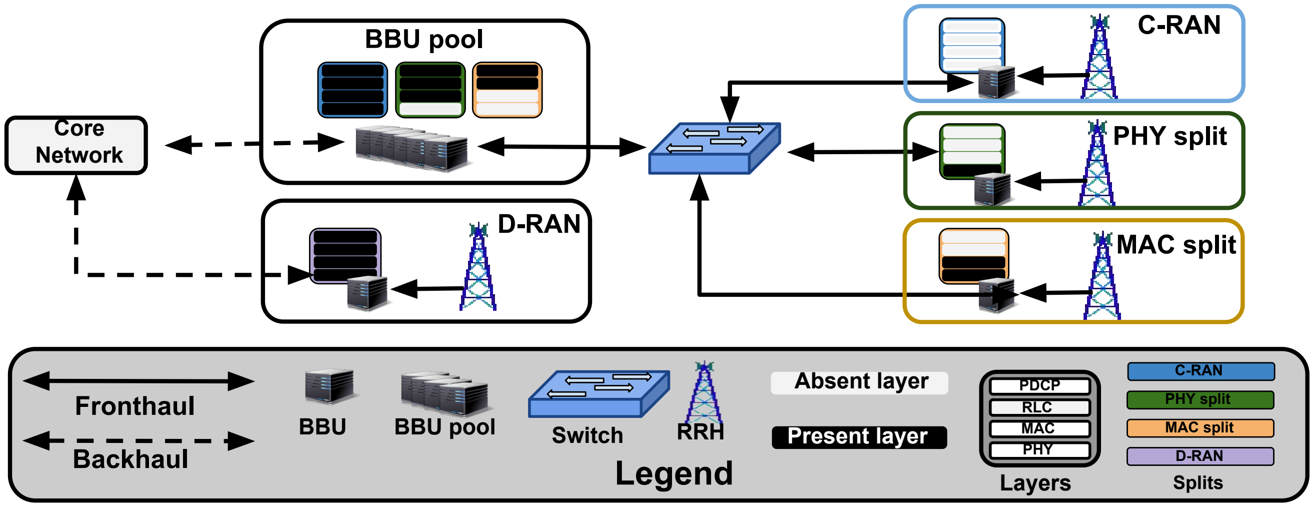


Fig. 2: The different functional splits that can co–exist at the BBU pool. Notice that, in the cases of the PHY and MAC splits, also the RRHs possess processing capabilities.

TABLE III: Virtual network request parameters

Variable	Description
G_v	Virtual network request.
N_v	Virtual nodes in G_s.
N_v^1	Virtual RRH sites in G_v.
N_v^2	Virtual BBU pools in G_v.
E_v	Virtual links in G_v.
$\omega_{ant}^v(n)$	RF front–ends required at RRH site$n \in N_v$.
$\omega_{prb}^v(n)$	PRBs required at RRH site $n \in N_v$.
$\omega_{prc}^v(n)$	Processing capacity required at RRH site $n \in N_v^1$.
$\omega_{bwt}^s(e^{nm})$	Capacity required for link $e^{nm} \in E_v$ (in Gbps).
$loc(n)$	Desired geographical location for RRH site $n \in N_v$.

virtual RRH site $n \in N_v^1$ can be placed from the preferred location specified by $loc(n)$. Table III summarizes the virtual network request parameters.

V. Virtual Network Embedding

Upon arrival of a new 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 first step (node embedding), each virtual node in the request is mapped to a substrate node. In the second step (link embedding), each link is mapped to a single substrate path. In both cases some constraints must be satisfied.

A. *ILP Formulation*

In order to properly map the location constraint, we need to modify the substrate network. Every virtual small cell $n' \in N_v^1$ in a request has a desired location $loc(n')$. Whereas, every substrate RRH node $n \in N_s^1$ has both location $loc(n)$ and a coverage radius $\delta(n)$. For each virtual RRH node $n' \in N_v^1$, we can then define a cluster of candidate RRH nodes $\Omega(n')$ to which the virtual small cell $n' \in N_v$ can be mapped:

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

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

$$\begin{aligned} min\Big(\sum_{n' \in N_v^1} \sum_{n \in \Omega(n')} \sum_{n^\star \in \Omega(n)} \sum_{p,p^\star \in \omega_{prb}^s(n) \times \omega_{prb}^s(n^\star)}^{p=p^\star} \Phi_p^n \Phi_p^{n^\star} \\ + \sum_{n' \in N_v^1} \sum_{n \in N_s} \sum_{m \in R_n} R_n(m) \Phi_{n,m}^{n'} \Big) \end{aligned} \tag{2}$$

where (with a slight abuse of notation) we use $n^\star \in \Omega(n)$ to indicate a node within radio coverage of the candidate substrate node $n \in N_s^1$ (i.e. an interfering node). Moreover, $\Phi_p^n, \Phi_{p^\star}^{n^\star} \in \{0,1\}$ are binary variables, indicating whether the PRBs $p, p^\star$ are in use at the substrate nodes n and $n^\star$ respectively.

The first term in the objective function aims at minimizing the number of overlapping PRBs while the second term minimizes the fronthaul bandwidth requirements. The rationale behind this approach is that different functional splits can enable different interference management techniques [29] and thus a trade–off exists between fronthaul bandwidth requirements and the level of acceptable interference in the system.

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

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

We will now detail the constraints used in the ILP formulation. The following constraints deal with the resources required to embed the requests, making sure that those resources are at most be equal to the resources available at the substrate nodes:

$$\sum_{n'\in N_v^1} \omega^v_{prc}(n')\Phi^{n'}_{n} \le \omega^s_{prc}(n) \quad \forall n \in N_s^1 \tag{4}$$

$$\sum_{n'\in N_v^2} \omega^v_{prc}(n')\Phi^{n'}_{n} \le \omega^s_{prc}(n) \quad \forall n \in N_s^2 \tag{5}$$

$$\sum_{e'\in E_v} \omega^v_{bwt}(e')\Phi^{e'}_{e} \le \omega^s_{bwt}(e) \quad \forall e \in E_s \tag{6}$$

$$\sum_{n'\in N_v^1} \omega^v_{ant}(n')\Phi^{n'}_{n} \le \omega^s_{ant}(n) \quad \forall n \in N_s^1 \tag{7}$$

$$\sum_{n'\in N_v^1} \omega^v_{prb}(n')\Phi^{n'}_{n} \le \omega^s_{prb}(n) \quad \forall n \in N_s^1 \tag{8}$$

where, $\Phi^{n'}_{n}, \Phi^{e'}_{e} \in \{0,1\}$ are binary variables indicating whether the virtual node $n' \in N_v$ and the virtual link $e' \in E_v$ have been mapped to the substrate node $n \in N_s$ and to the substrate link $e \in E_s$, respectively.

Each requested virtual small cell $n' \in N_v^1$ must be mapped only once (9) and only on a substrate node that belongs to its cluster of candidates (10):

$$\sum_{n\in N_s^1} \Phi^{n'}_{n} = 1 \quad \forall n' \in N_v^1 \tag{9}$$

$$\sum_{n\in N_s^1 \setminus \Omega(n')} \Phi^{n'}_{n} = 0 \quad \forall n' \in N_v^1 \tag{10}$$

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

$$\sum_{n'\in N_v^1} \Phi^{n'}_{n,p} \le 1 \quad \forall n \in N_s^1 \quad \forall p \in \omega^s_{prb}(n) \tag{11}$$

where $\Phi^{n'}_{n,p} \in \{0,1\}$ is a binary variable showing whether the PRB $p \in \omega^s_{prb}(n)$ of the substrate node $n \in N_s^1$ has been allocated to the virtual node $n' \in N_v^1$.

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

$$\sum_{p\in\omega^s_{prb}(n)} \Phi^{n'}_{n,p} - \omega^v_{prb}(n')\Phi^{n'}_{n} = 0 \quad \forall n' \in N_v^1 \quad \forall n \in \Omega(n') \tag{12}$$

In order to compute the fronthaul bandwidth requirement for the virtual small cell $n' \in N_v^1$, each virtual small cell has to be mapped to one and only one functional split:

$$\Phi^{n'}_{n} - \sum_{m\in R^n} \Phi^{n'}_{n,m} = 0 \quad \forall n' \in N_v^1 \quad \forall n \in \Omega(n') \tag{13}$$

where R^n is the set of the fronthaul bandwidths that would be required to support the request using different functional splits, and its index m indicates the functional split option. $\Phi^{n'}_{n,m}$ is a binary variable indicating whether the virtual small cell $n' \in N_v^1$ has been mapped to the m^{th} functional split option of substrate node $n \in \Omega(n')$.

The next constraint ensures that only one functional split is selected for a given substrate node. This constraints along with the previous one (13) make sure that all the virtual small cells, mapped to the same substrate node, have selected the same functional split:

$$\sum_{n'\in N_v} \Phi^{n'}_{n,m} - \sum_{n'\in N_v} \Phi^{n'}_{n} = 0 \quad \forall n \in N_s \quad \forall m \in R^n \tag{14}$$

The next constraint handles the functional split selection for each small cell:

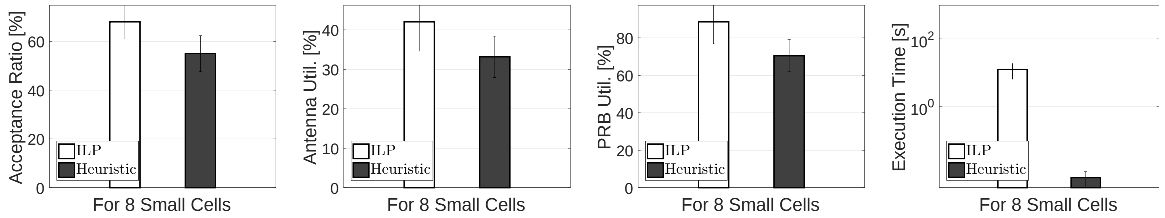
$$\sum_{n^\star\in\Omega(n)} \; \sum_{p,p^\star\in\omega^s_{prb}(n)\times\omega^s_{prb}(n^\star)}^{p=p^\star} \Phi^{n,n^\star}_{p,p^\star} \le I(m) \sum_{n^\star\in\Omega(n)} \omega^s_{prb}(n^\star)$$
$$\forall n' \in N_v^1 \quad \forall n \in \Omega(n') \quad \forall m \in R^n \tag{15}$$

where $I(m)$ represents the inter–cell interference level range for each $m \in R^n$ functional split option (see Table IV). This constrain effectively puts an upper bound on the number of acceptable overlapping PRB allocations. For example, for a C–RAN 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 in that a C–RAN split enables several advanced interference mitigation techniques. Conversely, as the functional split moves up in the protocol stack we reduce the maximum number of allowed overlaps in the PRBs allocation.

B. Heuristic

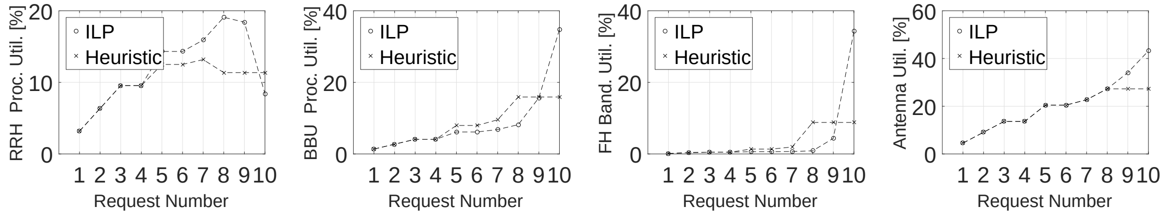
The ILP–based VNE algorithm becomes computationally intractable as the size of the substrate network and/or of the virtual network requests increases. For example, the ILP algorithm takes one week to map a request, having 20 virtual nodes, to a substrate network with 20 nodes. In order to address this scalability issue, we also put forward a heuristic algorithm, which is able to embed the same virtual network request in real time. The proposed heuristic algorithm is composed of three steps, implementing a joint node and link embedding. Let $m = |N_s^1|$ and $n = |N_v^1|$ be the number of, respectively, substrate and virtual RRHs, and $p = |R^n|$ be the number functional split options considered in this study. In the first step, a cluster of candidate RRHs are selected for each $n' \in N_v^1$ virtual RRH node, considering its requirements in terms of antenna configuration, number of PRBs and desired location. This process takes $O(nm)$ time. In the second step, all the candidates $m \in candidates(n')$ of each virtual RRH $n' \in N_v^1$ are considered, and for each of them, a neighbour list $neighbor(m)$ is created.

The list contains all the substrate RRHs whose downlink signal may interfere with the signal of $m \in candidates(n')$ candidate RRH. The relative distance between the potential interfering small cell is considered for populating the neighbour list. The algorithm then measures the interference coming from each of the small cells in the neighbour list. At the end of this step, the best candidate a small cell is picked. This process requires $O(nm^2)$ time. In the last step, the requested substrate resources are allocated, and the most appropriate functional



(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 of the ILP and the heuristic algorithms.



(a) RRH resource utilization. (b) BBU resource utilization. (c) Fronthaul band. utilization. (d) RF front–end utilization.

Fig. 4: Resource utilization using both the ILP and the heuristic placement algorithms.

Algorithm 1 Nodes and links assignment

```
procedure Input:($G_s$, $G_v$)
    Step 1: Compute a list of candidates.
    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
                if $\omega_a^v(n') \leq \omega_a^s(n)$ and $\omega_{prb}^v(n') \leq \omega_{prb}^s(n)$ 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.
        $min\_intf(n') \leftarrow \infty$
        for $m \in candidates(n')$ do
            for $n \in N_s^1$ do
                $d \leftarrow dis(loc(m), loc(n))$
                if $n \neq m$ and $d \leq 2\delta(n)$ then
                    $neighbor(m) \leftarrow n$
                end if
            end for
            $intf(n', m) \leftarrow 0$
            for $p \in neighbor(m)$ do
                for $q \in prb(p)$ do
                    if $p(q) = m(q) = 1$ then
                        $intf(n', m(q)) = intf(n', m(q)) + 1$
                    end if
                end for
            end for
            if $intf(m) \leq min\_intf(n')$ then
                $min\_intf(n') = intf(m)$
                $best\_candidate(n') \leftarrow m$
            end if
        end for
        Step 3: Allocate resources & select functional splits.
        $mapped(n') \leftarrow best\_candidate(n')$
        $sorted(m) \leftarrow sortprb(intf(m) \uparrow)$
        $alloc\_prb(n') \leftarrow sorted(m)[1 : rqst\_prb(n')]$
        for $s \in splits$ do
            if $intf(m) \in intf\_bounds(s)$ then
                $split(m) \leftarrow splits(s)$
            end if
        end for
        $fh\_band(n') \leftarrow compute\_band(\omega_{prb}^v(n'), \omega_a^v(n'))$
        Update substrate resources
    end for
end procedure
```

splits is selected for each substrate small cell that has hosted at least one virtual RRH. Thus, $n' \in N_v^1$ virtual RRH is mapped to the best substrate RRH $best\_candidate(n')$. Then the available PRBs of the host substrate small cell are sorted in the ascending order of likelihood in terms of interference, which is followed by the allocation of the requested PRBs starting from the least interfering PRB.

After allocating the requested number of PRBs, the overall interference level at the host small cell is estimated and the appropriate functional spit is selected, having an objective of minimizing the level of inter–cell interference at the minimum cost of fronthaul bandwidth consumption. Lastly, based on the selected split, the fronthaul bandwidth, which would be required to serve the embedded virtual small cell, is computed, and all the consumed substrate resources are updated. This final step requires $O(np)$ time. Thus, the overall time complexity of algorithm is $O(np + nm + nm^2) \approx O(nm^2)$.

VI. Evaluation

The goal of this section is to compare the ILP–based and the heuristic placement algorithms. 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 is a star–shaped network with 8 RRHs and a single BBU pool. The radio coverage of each RRH is 500 meters, providing a coverage in an area of 2 KM^2. Inter–RRH distance is 800 meters. This means that in some zones there will be 200 meters of area which will be covered by more than one RRH, which in turn means that, if the users are located in that area and are connected to different RRHs, being scheduled at the same PRBs, they will

TABLE IV: RRH and BBU relative processing capabilities and the interference range for the considered functional splits.

Split	$I(m)$	Processing Capacity	
		RRH	BBU
C-RAN	1	$0 \cdot \omega_c^v(n)$	$1 \cdot \omega_c^v(n)$
PHY split	0.6	$0.5 \cdot \omega_c^v(n)$	$0.5 \cdot \omega_c^v(n)$
MAC split	0.3	$0.7 \cdot \omega_c^v(n)$	$0.3 \cdot \omega_c^v(n)$

then create interference on one another, irrespective whether or not those users belong to the same virtual service provider (e.g., MVNOs). It is assumed that these RRHs are served by a single BBU pool, and 70% of the fronthaul links are direct optical fibers (10 Gbps), while remaining 30% of the fronthaul links are LoS millimeter wave (MMW) links (5Gbps). The RRHs using optical–fiber fronthaul links possess 8 RF front–ends, while the rest of the RRHs posses 4. This is a very conservative assumption, in more realistic scenario a ring or a bus topology could be used to connect the RRHs with the BBU pool. However, the focus of this study is on the flexible functional split, rather than fronthaul topology.

Depending upon the level of inter–cell interference, different functional splits would be appropriate to be exploited. In our model, there are three categories of interference and three corresponding functional splits (see Table IV). Notice that as opposed to the C–RAN case, in the case of Split PHY and Split MAC, the RRHs do possess processing capacity. For example, in the case of Split PHY, it is assumed that the half of the processing capacity is allocated to the RRHs and the other half to the BBU pools. 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 RRHs and decreases at the BBU pools when a fewer layers (e.g., PHY layer, MAC layer) are centralized at the BBU 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. During each embedding the number of virtual RRHs, RF front–ends and PRBs are randomly selected in the set of $\{1, 2\}, \{1, 2\}$ and $\{25, 50\}$, respectively.

B. Simulation Results

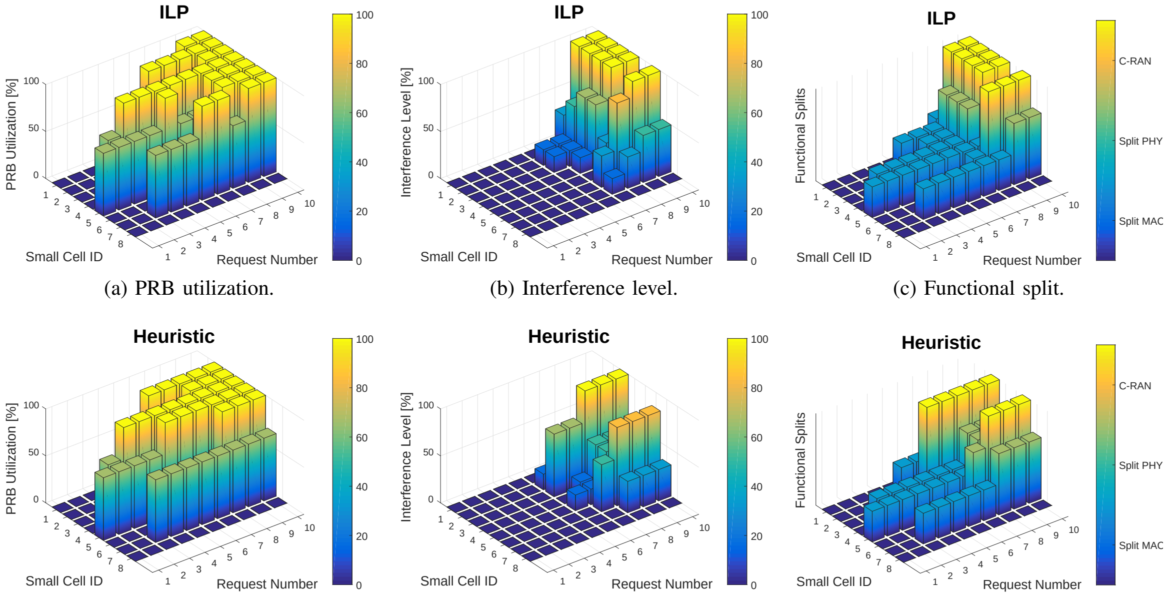
Figure 3 shows the acceptance ratio, the RF front–end and PRB utilization, and the execution time of the ILP–based and the heuristic algorithms. As expected, the ILP–based algorithm has better performance in terms of accepted requests (see Fig. 3a). For this reason, the network–wide antenna utilization as well as the PRB utilization of the ILP–based algorithm are higher than the ones of the heuristic (see Fig. 3b and Fig. 3c). Unfortunately, the ILP algorithm becomes computationally intractable when large networks with a few tens of small cells are considered. Indeed, as the Fig. 3d displays, the average time taken to embed a single request to the substrate network for the heuristic algorithm is at least two orders of magnitude less compared to the one of the ILP algorithm.

In order to get a better insight into how the processing, fronthaul bandwidth and RF front–end resources of the substrate network are exploited, we will now examine a single iteration (i.e., 10 embeddings). Figure 4 depicts the processing resource utilization of the RRHs and the BBU pool, the overall fronthaul bandwidth utilization and the RF front–end utilization of both the ILP–based and the heuristic algorithms. It is important to mention that the embedding algorithm is dynamic, meaning that with the arrival of a new embedding request, all previously embedded requests along with the new one are re–embedded, therefore, finding the globally optimal embedding solution for each request.

It can be observed that, for the ILP algorithm, the processing resource utilization of the RRHs keeps increasing until the 8^{th} embedding (see Fig. 4a), reaching its saturating point in terms of the overall inter–cell interference level in the network. This means that, for 8 embeddings, the ILP employs the same lower layer (i.e., split MAC) functional split, since the interference level is within the acceptable range for that split option. Whereas, starting from the 9^{th} embedding, the processing resource utilization of the RRHs decreases, since higher layer (i.e., split PHY) functional splits are used, enabling more complex inter–cell interference reduction/cancellation algorithms to be employed. It can also be observed that, for the heuristic algorithm, the functional split changes from the 8^{th} embedding. This is because the interference level at the 8^{th} embedding exceeds the acceptable range for the lower layer split (i.e., split MAC), and split changes in order to apply inter–cell interference suppression algorithms.

The picture is totally different in the processing resource utilization of the BBU pool (see Fig. 4b), where the utilization increases with the number of embeddings. This is justified by the fact that, depending upon the interference level when more requests are embedded, the functional splits of the small cells change from the lower layer split towards the upper layer splits, and with this change, the processing resource requirement grows (see Table IV). It can be noticed that the processing resource utilization of the ILP algorithm is higher than the one of the heuristic algorithm. This is because the ILP–based algorithm has embedded more requests then the heuristic. Similarly, in Fig. 4c, the fonthaul bandwidth utilization of the ILP algorithm is more since the fronthaul bandwidth requirement increases with the change in the functional splits towards higher layers. Fig. 4d displays the RF front–end utilization of both algorithms. It can be seen that the algorithms embedded an equal number of requests until the 8^{th} embedding. It can also be seen that the heuristic algorithm failed to embed the 9^{th} and the 10^{th} requests, while the ILP–based algorithm embedded them successfully. Therefore, the network–wide RF front–end utilization of the ILP–based algorithm is higher than the one of the heuristic.

We will now examine the PRB utilization, interference level and the functional split for both algorithms for a single iteration in order to better understand their relationship (see Fig. 5). It can be noticed that the PRB utilization of the ILP algorithm is higher across the entire substrate network. This is justified by the fact that the ILP–based algorithm has embedded more requests than the heuristic. For the same



(a) PRB utilization.

(b) Interference level.

(c) Functional split.

(d) PRB utilization.

(e) Interference level.

(f) Functional split.

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

reason, also the ultimate inter–cell interference level of the ILP algorithm is higher. However, it is interesting to note that the picture initially is entirely different. Until the 5^{th} embedding, the inter–cell interference is absent in the substrate network, regardless of the employed algorithm. This means that both algorithms have been able to embed the first five requests and allocate the requested number of PRBs in such a way as to avoid introducing inter–cell interference into the network. Nevertheless, with increasing the number of embedded requests, the inter–cell interference becomes inevitable. In Fig. 5b and Fig. 5e, starting from the 6^{th} embedding, it can be observed that the inter–cell interference is introduced and its level grows with an increase in the number of embeddings. However, it is worthwhile to note that, for example, in the 6^{th} embedding when the ILP algorithm is employed, the inter–cell interference level in the substrate network is lower than the one of the heuristic. Moreover, in the ILP case, the inter–cell interference exist at two substrate small cells (i.e., small cell number 2 and 3), while in the heuristic case, the inter–cell interference exist at three substrate small cells (i.e., small cell number $2, 3$ and 5). This is because the ILP algorithm, although takes more time, is more efficient in embedding the incoming requests.

Finally, Fig. 5c and Fig. 5f show the functional splits per substrate small cell. The superiority of the ILP algorithm can be observed also here since, for example, if we consider the first eight embeddings that both algorithms have successfully embedded, we can see that the ILP uses more lower layer MAC split than its heuristic counterpart. This is because the ILP has embedded those requests more efficiently than the heuristic, resulting in less level of inter–cell interference. It can also be observed that, depending upon the inter–cell interference level, the functional split changes from the split MAC towards split PHY and split C–RAN. Thus, the lower is the inter–cell interference level, the lower layer split is selected, leading to more efficient utilization of the fronthaul bandwidth, therefore, minimizing the energy consumption across the InPs' network. Lastly, it can be noticed that at the end of all embeddings, more substrate nodes use C–RAN functional split in case of the ILP algorithm. This is a result of embedding more requests compared to the heuristic algorithm.

VII. Conclusions

Flexible functional split in the RAN provides the possibility of exploiting complex CoMP algorithms designed to reduce/cancel the inter–cell interference. However, depending upon the level of inter–cell interference, a particular functional split would be more efficient to be used. We have seen that the processing requirements of the RRHs and the BBU 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 lower layer PHY split toward upper layer MAC and C–RAN splits, 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 a commercial LTE+ mobile network in which both wireless and optical links are used as transport mediums. Based on the available of the transmission links as well as the spatially and temporarily fluctuating traffic demand at eNodeBs, 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] "C-RAN, The Road Towards Green RAN," China Mobile, Tech. Rep., 2011.

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

[4] F. Richter, A. J. Fehske, and G. P. Fettweis, "Energy efficiency aspects of base station deployment strategies for cellular networks," in *Proc. of IEEE VTC 2009-Fall*, Anchorage, AK, USA, 2009.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] "White paper of next generation fronthaul interface," China Mobile, Tech. Rep., 2015.

[14] "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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] M. Jaber and D. Owens and M. A. Imran and 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.

[24] C. Y. Chang and R. Schiavi and N. Nikaein and T. Spyropoulos and C. Bonnet, "Impact of packetization and functional split on C-RAN fronthaul performance," in *Proc. of IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[25] "Small cell virtualization functional splits and use cases," Small Cell Forum, Tech. Rep., June 2015.

[26] Zaki, Yasir and Zhao, Liang and Goerg, Carmelita and Timm-Giel, Andreas, "LTE wireless virtualization and spectrum management," in *Proc. of IEEE WCNC*, Budapest, Hungary, 2010.

[27] 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.

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

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