Optimization of an integrated fronthaul/backhaul network under path and delay constraints

Nuria Molner[a,b,*], Antonio de la Oliva[b], Ioannis Stavrakakis[a,b,c], Arturo Azcorra[a,b]

[a] *IMDEA Networks Institute*
[b] *Universidad Carlos III de Madrid*
[c] *National and Kapodistrian University of Athens*

Abstract

Cloud or Centralized Radio Access Networks (C-RANs) are expected to be widely deployed under 5G in order to support the anticipated increased traffic demands and reduce costs. Under C-RAN, the radio elements (e.g., eNB or gNB in 5G) are split into a basic radio part (Distributed Unit, DU), and a pool-able base band processing part (Central Unit, CU). This functional split results in high bandwidth and delay constrained traffic flows between DUs and CUs (referred to as fronthaul), calling for the deployment of a specialized network to accommodate them or for integrating them with the rest of the flows (referred to as backhaul) over the existing infrastructure. This work studies the next generation of transport networks, which aims at integrating fronthaul and backhaul traffic over the same transport stratum. An optimization framework for routing and resource placement is developed, taking into account delay, capacity and path constraints, maximizing the degree of DU deployment while minimizing the supporting CUs. The framework and the developed heuristics (to reduce the computational complexity) are validated and applied to both small and large- scale (production-level) networks. They can be useful to network operators for both network planning as well as network operation adjusting their (virtualized) infrastructure dynamically.

Keywords: 5G-Crosshaul, fronthaul, backhaul, optimization, delay

*Corresponding author

1. Introduction and Motivation

According to recent predictions [1] mobile data traffic will increase 11-fold between 2016 and 2021. In order to serve this increasing user demand in an environment of reduced revenues per user, a new generation of network designs is required, the so-called Fifth Generation of network architectures (5G). 5G will be characterized by an increased available bandwidth to the users, providing the user with unprecedented speeds, fostering the evolution and deployment of new services which were not possible before. In addition, to increase the available resources per area unit, it is expected that 5G deployments will feature a higher capillarity, effectively increasing the density of the network. Through this densification, spectrum can be reused in a more effective way, paving the way towards higher bandwidths available to the end user as foreseen by the Cooper's law[1].

One key element to support the increased bandwidth to the user is the transport network that feeds the Radio Access Network (RAN). The future 5G RAN must support an unprecedented amount of traffic, with very stringent requirements in terms of latency and jitter. This will heavily impact on the design of the transport network feeding the RAN that must support more demanding transport requirements. In addition, RAN designers are looking for innovative ways of improving the performance achievable by the RAN. One of the mechanisms already identified in the literature is to split the radio elements (e.g., eNB or gNB in 5G) into a small footprint basic radio part (Distributed Unit, DU), which may include lowest levels of the protocol stack, and a pool-able base band processing part (Central Unit, CU). This technology, known as Cloud or Centralized RAN (C-RAN), will be massively used in 5G since it helps reduce the costs associated with the RAN and provide an additional performance gain due to the pooling of resources and the coordinated processing of signals from different cells. The disadvantage of the C-RAN technology is the need for a high bandwidth and low delay network connection between the radio and processing parts. This network segment has traditionally been known as fronthaul and has recently been the subject of a lot of research on protocols (CPRI [2], eCPRI [3]) and analysis of the possible functional splits of the protocol stack [4, 5].

In this context, the operator faces a very complex and challenging network to manage, which is no longer divided into RAN, transport and core domains

[1] http://www.arraycomm.com/technology/coopers-law/

but places different RAN and core elements within data-centers distributed in the transport network. This new network, which is being referred to as Crosshaul [6], encompasses the front- and back-hauling network segments and requires new approaches for the planning and operation of the network. Operators now need to decide not only on the placement of each radio node but also whether it should be split, where the higher layers of such a split should be placed and how the resulting traffic sources affect the rest of the links.

This paper tackles a new methodology for the planning and operational optimization of the network, focusing on the integrated transport of fronthaul and backhaul traffic, the placement of the pool-able resources containing the radio nodes' higher layers (CUs) and the overall delay achievable in the network. In Section 2 we provide some background related to our problem. Section 3 presents a mathematical formulation that maximizes the DU deployment and yields the optimal number of data-centers containing the pool-able CUs and their location, while taking into account the stringent delay requirements of the resulting fronthaul traffic by incorporating proper queuing models. The general formulation is non-convex and non-linear and since non-tractable (as shown in Section Appendix A), certain approximations are also introduced in Section 3 to yield a tractable formulation that can provide with reasonable computational complexity for the optimal results, at least for the case of small scale environments. For larger-scale environments, a computationally tractable heuristic is introduced in Section 3.3 that provides for an efficient (though not necessarily optimal) solution, achieved in reduced time. In Section 4 the developed approaches are validated and applied to both small - and large - scale (production) networks and some results are presented. Finally, some conclusions are drawn in Section 5.

2. Background

The Crosshaul concept results from the convergence of different concepts that have been incorporated in cellular networks over the last years. First, the Centralized or Cloud-Radio Access Network (C-RAN) paradigm has emerged as a significant trend in mobile networks in order to reduce CAPEX and OPEX while increasing RAN's performance. Second, the evolution of C-RAN technologies from a serial transmission (CPRI) to fully packetized protocols (eCPRI) allows for the integration of fronthaul and backhaul networks. Finally, the deployment of intelligence at the edge of the network - in the form

of micro data-centers which can host virtual network functions including C-RAN baseband processing - has transformed the network from a mere data pipe into a smart application hosting environment.

Current 4G networks have started deploying the C-RAN concept in current networks. The seminal paper [7] proposes the use of fronthaul as a mechanism to reduce the CAPEX and OPEX due to reduced expenses on the site antenna. Later, fronthaul has been also proven useful to improve the performance of the air interface due to the easiness of synchronization of the Central Units, allowing the use of CoMP. The main problem with current C-RAN approaches is the use of CPRI (the predominant fronthaul technology) which uses a serial transmission, not encapsulated, requiring of point to point high bandwidth and dedicated fibers between the DUs and CUs. This increases the cost of management and operation of the network, since the operator now has to face the operation of two different networks, one based on packets (the normal backhaul or transport network) and a second one using a completely different technology. This fact has trigger a change in how standardization bodies has focused on C-RAN for 4G and 5G, working on solutions based on packets that can use standard switching technologies.

In the following we present some background information on each of these technologies, as well as related works on the optimization of the resulting converged network. The deployment of the 5G RAN is expected to capitalize on the concept of Centralized or Cloud-Radio Access Network (C-RAN) [8]. In C-RAN systems the base station functionality is split at a certain point of the protocol stack (such as, for example, the physical layer), and the upper part is moved to a central unit, typically within or co-located with an edge data center facility [9, 10]. A C-RAN system consists at least of the following three main components: the distributed units implementing the radio functions, the central processing units which are typically aggregated in pools, and the network interconnecting them, typically referred to as fronthaul [11]. The different points in the protocol stack that determine the separation of the functions processed in central or distributed units define what is referred to as *functional splits* [12, 4]. The implementation of a given functional split uniquely defines the properties of the system design [5]. The complexity, benefits and drawbacks of the distributed and centralized units depend on the functional split chosen. Currently, the most common functional split corresponds to the division at the physical layer (as implemented by the Common Public Radio Interface, CPRI). CPRI is a non-packetized serial protocol which cannot be integrated with other packetized transmis-

sions unless a circuit (e.g., a wavelength) is reserved for it. Hence in this work we consider the recently published evolution of CPRI (eCPRI), which packetizes the I/Q samples in an Ethernet compatible way.

The C-RAN approach is to benefit significantly through virtualization. By virtualizing and centralizing the baseband processing of multiple cells, an operator is able to better manage inter-cell interference and traffic load, as well as reduce overall costs. At the same time, by co-locating multiple centralized units pooling gains appear and scaling up the system when RAN demands increase is facilitated. The current trend towards the deployment of Edge data centers, aiming at hosting delay constrained applications - such as augmented reality - has opened the door for deploying C-RAN centralized units in virtualized infrastructure at the edge of the network.

The efficient design and operation of such an environment requires a joint consideration of routing, placement of Edge data center and C-RAN cell deployment in the presence of traffic with multiple priorities and strict deadlines and, thus, extending the state of the art. Work related to delay constrained routing may be found in [13]. This work proposes a Djikstra shortest path algorithm that uses link delay as the weight of a link. In [14], the authors propose a heuristic algorithm based on the minimum delay path and shortest path for networks with time-dependent edge-lengths. Heuristic algorithms to derive the minimum cost (delay) tree between the source and the destination can also be found in the work in [15]. Routing with delay constraints and analysis of delay variation has also been studied for multicast networks in [16]. The M/G/1 queuing model is one widely adopted for modeling the queuing delay in network nodes. For instance, in [17], the authors introduce fixed parameters for the arrival rates and exit rates (λ and μ) in order to make the problem tractable. In [18, 19, 20, 21] the authors approximate the delay with non linear equations. In [22] authors deal with the M/G/1 queueing model with priorities for the problem of the mixed fronthaul and backhaul networks proving that it is a good approximation for this traffic. Authors in [23] used M/M/1 queueing model for Virtual Network Function (VNF) placement problems without several priorities and dealing with non-linear equations. Finally, works on un-splittable flow problems, such as [24], [25] and [26], develop heuristic algorithms for NP-hard problems in the general case, but these papers do not consider networks integrating traffic with different priorities.

The unified problem considered in this paper also addresses the optimal placement of the Edge data centers, further enhancing the applicability

and complexity of the work. This problem is related to the problem of the Virtual Network Embedding (VNE) and the problem of placing chains of virtual functions [27]. In our paper, we consider the problem of determining the optimal placement of the data centers subject to the transformation of fronthaul flows (with special characteristics) into backhaul flows. Finally, work on transporting fronthaul traffic over IEEE 802.1Q switches has also been recently carried out. Works such as [28] conclude that such a transport is possible through the extension provided by 802.1Qbu, 802.1Qbv and by employing buffers at the receivers.

Although the past work - as the above - shows that RAN centralization has many advantages and has been tackled through multiple perspectives, this paper, to the best of our knowledge, is the first to study the complete problem of the joint optimization of C-RAN deployment and Edge data center placement, taking into consideration the stringent fronthaul flow deadlines and the accumulated delay in the switching devices.

3. General Problem Statement and Optimization Formulation

Figure 1 depicts the general environment considered in this paper, where a number of sources are connected to the Internet (where the potential destination of a source flow is assumed to reside) through an edge/cloud access network. The sources represent either the base station of a classical RAN (e.g. an eNB node) or just a Distributed Unit (DU) of a C-RAN whose upper layer functions are executed somewhere in the Edge/Cloud network by the Central Units (CUs). The traffic flow departing a DU source (fronthaul) is typically of high rate (e.g., 0.9Gbps), although its exact bandwidth depends on the functional split used and the channel bandwidth use (among others, depending on the functional split). On the other hand, the traffic flows departing an eNB node (backhaul) are of much lower rate, can be several and up to a maximum total rate of typically 0.15Gbps under full utilization of the air medium of the eNB node (depending on MIMO and bandwidth configuration of the eNB). This topology of mixed RAN and C-RAN components is expected to dominate for the foreseen future, as a progressive migration from a RAN to a C-RAN dominated world takes place for the benefits discussed earlier.

The main objective in this paper is to provide for efficient or optimal designs of such mixed RAN/C-RAN environments. These mixed environments emerge as operators attempt to maximize their adoption of the C-RAN tech-

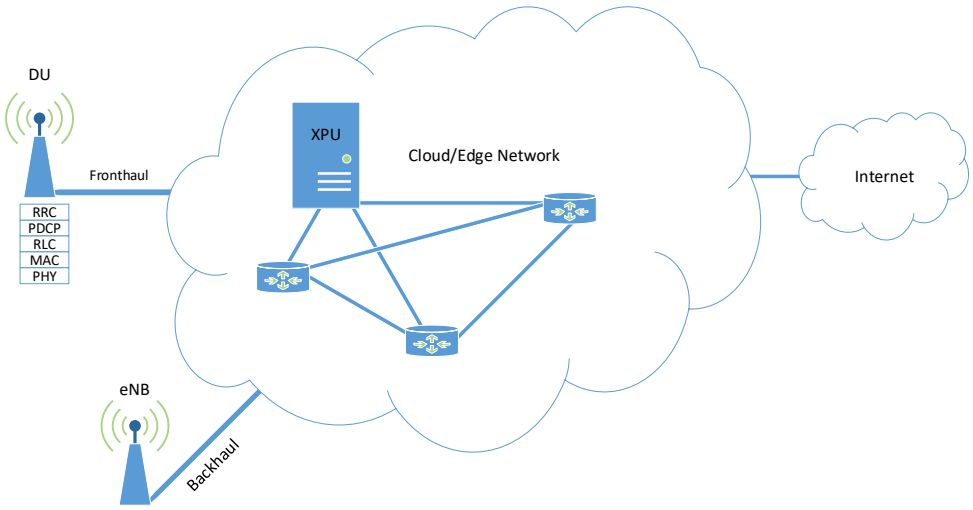


Figure 1: The general network environment

nology in the most effective way, subject to the constraints imposed by the available supporting infrastructure (explained below). Or, such environments may emerge in a more dynamic (operation-level) case, where operators may switch on or off Distributed Units or aggregate them in a lower number of Central Units according to the demand to reduce OPEX; re-optimization of the resulting mixed RAN / C-RAN environment is then needed as well.

To facilitate the discussion on the formulation of the optimization problem in this section, an originally all RAN environment will be considered and seek to optimize the degree of migration towards a mixed RAN/C-RAN environment by minimizing the number of (remaining) RAN components and optimizing the C-RAN deployment in the resulting environment. That is, maximize the number of RAN components that are replaced by a DU, while maximizing the degree of aggregation/pooling of the CU components by minimizing the number of locations hosting the CUs. The latter pooling provides for some wireless capacity enhancement through coordinated signal processing and reduces costs for the operator. All the co-located CUs will be considered as components of a single data-center, to be referred to as Crosshaul Processing Unit (XPU). The capacity of an XPU is considered to be equal to the number of CUs available/implemented in the specific location.

The main challenge in this migration is due to the fact that the (single) flow departing a DU (referred to a fronthaul): (a) has a (much) higher rate compared to that of the original eNB; (b) it must be routed towards a XPU (containing a CU) facility to be processed first, before it is transformed into backhaul flow(s) and be routed from there towards the destination; (c) it has stringent delay requirements for reaching the CU facility. In addition, the location of the CU facility - where a fronthaul flow is forwarded to - needs to be determined by minimizing the number of such locations (maximizing CU pooling), or minimizing the number of XPU facilities deployed. These challenges are incorporated and addressed through the optimization formulation developed and solved in this paper.

As indicated earlier, this paper considers a single type of packetized fronthaul flow (eCPRI), although the model presented could be used for any other kind of packetized fronthaul. This flow exists only over the path between the generating DU and the associated XPU and it becomes a standard backhaul flow over the path between the XPU and its destination. Such standard backhaul flows are also generated by the classical RAN nodes (e.g., eNBs) and will coexist with fronthaul flows. The key parameters of the two types of flows that are considered in this paper are given in Table 1 from [2] and [29]. A quick comparison of the two flows shows that the fronthaul flow is of much higher rate and has much less delay requirements. This large asymmetry in these parameters leads to a number of observations and design considerations.

As a fronthaul flow has very stringent delay requirements (compared to a backhaul flow) it will be treated as a class of traffic of (non-preemptive) priority 1. A severe consequence of changing an eNB node to a DU+CU is that a high rate increase will be observed in the links departing the eNB/DU node towards the location of the associated CU (XPU to be determined). This severe increase in the rates along with the stringent delay requirement over that part of the network constraint the migration from a RAN to a fully C-RAN environment. Finally, the high rate asymmetry makes the approximation of considering in our optimization formulation a single (as opposed to multiple) backhaul flow departing the associated CU a reasonable one, as it is expected to have only minor impact on the solution of the optimization problem, which is primarily shaped by the pre-XPU part of the access network.

The notation for the various parameters employed in the formulation of the optimization are shown in Table 2. Notice that a fronthaul source

Flow	Value	Delay	Class
fronthaul (eCPRI)	900 Mbps	250 μs	1
backhaul (fronthaul after XPU usage)	150 Mbps	100 ms	2
backhaul	15 Mbps	100 ms	2

Table 1: Parameters of fronthaul and backhaul traffic considered

generates one flow, while a backhaul source generates up to K flows, denoted by k, $0 \leq k \leq K$. It is also worth to mention that the work performed in this article assumes values for the air interface in line with 4G deployments, since right now there are no deployments of 5G C-RAN or even 5G air interfaces. However, the moment 5G is deployed, operators will need to enhance their transport networks, that now need to transport much more capacity to the RAN. In that moment, these values may differ from the ones we chose but this is just a parameter of the model, which can be easily changed and simulations re-run, yielding to different results in terms of total air capacity or number of XPUs. Therefore, the operators will only have to perform the simulations with the new values, but the main contribution, the mathematical model, heuristics and tendencies on the results will be the same.

$\mathcal{F}$	set of sources
f^l	rate of fronthaul source/flow l, $l \in \mathcal{F}$ (Mbps)
f^{l+}	rate of fronthaul flow l, $l \in \mathcal{F}$, after using its CU (Mbps)
b_k^l	rate of flow k of backhaul source l, $l \in \mathcal{F}$ (Mbps)
p^l	packet size of fronthaul flow l (0.012 Mbits=1500bytes)
p_k^l	packet size of flow k of backhaul source l (0.012 Mbits)
D^{l-}	delay constraint of fronthaul flow l, to reach its CU
D^l	delay constraint of fronthaul flow l
D_k^l	delay constraint of flow k of backhaul source l
$\mathcal{E}$	set of links of access/edge network
c_{ij}	capacity of link (i,j) (Mbps)
μ_{ij}	capacity of link (i,j) (in packets/sec)
$L_{i,j}$	length of link (i,j)
$\mathcal{X}$	set of available XPU facilities
N_r	capacity of XPU r, $r \in \mathcal{X}$ (in CUs)

Table 2: Parameters employed in the formulation of the optimization

Let $\mathbb{I}^{*}_{**}$ denote a binary variable (b.v.) assuming the value 1 if an event specified through ** and * has occurred, and 0 otherwise. The main binary and other variables employed in the formulation of the optimization problem are defined in Table 3.

$\mathbb{I}^{l}$	b.v. indicating if source (flow) l is a DU (fronthaul), $l \in \mathcal{F}$ (if not, it is an eNB)
$\mathbb{I}^{l}_{ij}$	b.v. indicating if link (i,j) is used by fronthaul flow l, before reaching its CU, $l \in \mathcal{F}$
$\mathbb{I}^{l+}_{ij}$	b.v. indicating if link (i,j) is used by fronthaul flow l, after leaving its CU, $l \in \mathcal{F}$, (and has then been transformed into a backhaul flow)
$\mathbb{I}^{l}_{ijk}$	b.v. indicating if link (i,j) is used by flow k of backhaul source l, $l \in \mathcal{F}$
$\mathbb{I}^{XPU}_{r}$	b.v. indicating if XPU r is used
$\mathbb{I}^{XPU,l}_{r}$	b.v. indicating if XPU r is used by flow l, $l \in \mathcal{F}$
λ^{n}_{ij}	rate of priority class n entering link (i,j) (in packets/sec)
d^{l}	delay of fronthaul flow l, $l \in \mathcal{F}$
d^{l-}	delay of fronthaul flow l, $l \in \mathcal{F}$, until it reaches its CU
d^{l}_{k}	delay of flow k of backhaul source l, $l \in \mathcal{F}$

Table 3: List of main binary variables (b.v.) and other variables employed in the formulation of the optimization.

3.1. Formulation of the optimization problem

In this subsection we employ the notation presented above to formulate the optimization problem by presenting the objectives, the various traffic and resource constraints and the supporting equations. The treatment of the delay constraints is deferred to the next subsection.

A set of locations of the sources of traffic are given (whose type fronthaul/backhaul is to be determined), along with the Edge/Cloud network topology (link capacities and lengths), the set of network nodes which are capable of hosting an XPU facility and the maximum number of XPUs to be possibly deployed. The solution of the optimization problem will determine the type of each one of the sources, while maximizing the number of fronthaul traffic sources and minimizing the number of XPUs deployed whose location is also determined. As discussed earlier a source can be accepted as

a fronthaul source only if it is supported by a non-dedicated XPU, to yield some pooling gain; that is, if its CU can be hosted in an XPU (location) that supports at least one more CU serving another fronthaul source. The available link capacity between the sources and the XPU location will be the constraining factor determining whether CU collocation is possible or not.

To ensure that the maximum number of DU sources is determined under the constraint that all of them are supported by non-dedicated XPUs, the following objective function is defined for some $g > 1$.

$$\max \left\{ g \cdot \sum_{l} \mathbb{I}^{l} - \sum_{r} \mathbb{I}_{r}^{XPU} \right\} \tag{1}$$

Notice that the above objective function prescribes the following gains or penalties: (a) maximizes the number of DUs in the network, (b) minimizes the number of XPUs that are used for those DUs, and (c) if a source is an eNB source, it is not associated with any XPU and, thus, it does not contribute to the objective function (its gain is zero). Notice that, based on the above, an eNB source is preferable over a DU source supported by a dedicated XPU, because in the constraint (10) we impose that the XPUs cannot be dedicated to one DU. Similarly, a DU supported by a non-dedicated XPU is preferable over (yields a higher gain than) an eNB source. Consequently, the objective in Equation (1) ensures that the solution to the maximization will not contain any DU source that is supported by a dedicated XPU, the number of DUs will be maximized and the number of XPUs will be minimized. The latter is the case since it can be easily shown than the resulting gain associated with M_1 non-dedicated XPUs is higher than that associated with M_2 non-dedicated XPUs for $M_1 < M_2$, for the same number of DUs.

In order to accommodate the various resource constraints and other specific requirements, various equations and constraints are introduced next and summarized in Table 4.

The requirement for single path routing implies that all the traffic of any source leaves the source through a single link, as captured by Equations (2) and (3).

$$\mathbb{I}^{l} = \sum_{j} \mathbb{I}_{lj}^{l} \quad \forall\, l \in \mathcal{F} \quad \forall (l,j) \in \mathcal{E} \tag{2}$$

$$(1 - \mathbb{I}^{l}) = \sum_{j} \mathbb{I}_{ljk}^{l} \quad \forall\ flow\ k\ of\ source\ l \in \mathcal{F} \tag{3}$$

Constraint	Function
Source Constraints Eqs. (2) and (3)	To determine if a source is an eNB or a RU
Link Capacity Eq. (4)	To assure the traffic that uses a link does not surpass its capacity
Destination Constraint Eq. (5)	To assure all the traffic reaches its destination
XPU Constraints Eqs. (6) - (17)	To assure the fronthaul flows are processed in a XPU
Node Constraints Eqs. (20) - (22)	To assure non loss of traffic
Single Path Eqs. (23) - (25)	To assure single path for all the flows in the network
Delay Constraints Eqs. (26) - (39)	To compute the delay of the traffic

Table 4: Summary of main constraints of the optimization.

The link capacity constraints are captured by Equation (4).

$$\sum_{l} f^{l} \cdot \mathbb{I}_{ij}^{l} + \sum_{l} f^{l+} \cdot \mathbb{I}_{ij}^{l+} + \sum_{l,k} b_{k}^{l} \cdot \mathbb{I}_{ijk}^{l} \leq c_{ij} \quad \forall (i,j) \in \mathcal{E} \tag{4}$$

As the design space in this paper is the Edge/Cloud network and the deployed C-RAN, it is assumed that the destination of each flow is beyond this network and is referred to as "the Internet"; let the superindex in j mark a (fictitious) node representing that Internet destination of the flow. Notice also that a fronthaul flow always reaches its (Internet) destination as a backhaul flow. Equation (5) captures the balance of flows entering and exiting the edge/access network.

$$\sum_{l,k} \mathbb{I}_{k}^{l} + \sum_{l} \mathbb{I}^{l} = \sum_{i,j^{Int},l,k} \mathbb{I}_{ij^{Int}k}^{l} + \sum_{i,j^{Int},l} \mathbb{I}_{ij^{Int}}^{l+} \tag{5}$$

Equation (6) captures the requirement that a fronthaul flow must be routed through a node hosting an XPU.

$$\mathbb{I}^{l} = \sum_{r} \mathbb{I}_{r}^{XPU,l} \quad \forall\, l \in \mathcal{F} \tag{6}$$

Equation (7) imposes the requirement that a fronthaul flow l, $l \in \mathcal{F}$, that uses XPU r, $r \in \mathcal{X}$, must use one (and only one) of the incoming links attached to the node hosting the XPU (referred to as j^{XPU_r}); if flow l does not use this XPU, it may still use one of its incoming links.

$$\mathbb{I}_r^{XPU,l} \leq \sum_i \mathbb{I}_{ij^{XPU_r}}^l \quad \forall r \in \mathcal{X}, \quad \forall\, l \in \mathcal{F} \tag{7}$$

Equation (8) captures the capacity constraint of an XPU r

$$\sum_l \mathbb{I}_r^{XPU,l} \leq N_r \quad \forall r \in \mathcal{X} \tag{8}$$

An XPU is considered to be utilized as long as at least one fronthaul source uses it. On the other hand, an XPU has to be used by at least two fronthaul sources. These constraints are captured by Equations (9) and (10).

$$\mathbb{I}_r^{XPU,l} \leq \mathbb{I}_r^{XPU} \quad \forall r \in \mathcal{X} \tag{9}$$

$$2 \cdot \mathbb{I}_r^{XPU} \leq \sum_l \mathbb{I}_r^{XPU,l} \quad \forall r \in \mathcal{X} \tag{10}$$

A fronthaul flow l entering node j^{XPU_r} hosting XPU r, will appear at an outgoing link as either a transformed backhaul flow if it is processed by XPU r, or as the same fronthaul source otherwise; this is captured by Equation (11). A fronthaul flow l that has been transformed into a backhaul flow (having been processed by another XPU) entering node j^{XPU_r} hosting XPU r, will appear unmodified at an outgoing link; this is captured by Equation (12).

$$\begin{aligned}\sum_i \mathbb{I}_{ij^{XPU_r}}^l = &\left(\sum_i \mathbb{I}_{j^{XPU_r}i}^{l+}\right) \cdot \mathbb{I}_r^{XPU,l} + \\ &+ \left(\sum_i \mathbb{I}_{j^{XPU_r}i}^{l}\right) \cdot \left(1 - \mathbb{I}_r^{XPU,l}\right)\end{aligned} \tag{11}$$

$$\sum_i \mathbb{I}_{ij^{XPU_r}}^{l+} = \left(\sum_i \mathbb{I}_{j^{XPU_r}i}^{l+}\right) \cdot \left(1 - \mathbb{I}_r^{XPU,l}\right) \tag{12}$$

Notice that the above constraints are non-linear and would increase the complexity of the optimization machinery to be employed. As we aim at keeping the computational complexity low and use linear programming tools, we linearize these constraints as described below. To facilitate the presentation, we use the notation shown below for the two terms in the right hand side of Equation (11), which is rewritten as in Equation (13)

$$\sum_i \mathbb{I}^l_{ij^{XPU_r}} = \alpha_{rf^{l+}} + \alpha_{rf^l} \tag{13}$$

where $\alpha_{rf^{l+}}$ is bounded from above and below by the linear expressions shown in Equations (14) and (15) and α_{rf^l} is bounded from above and below by Equations (16) and (17). Notice that it is possible to linearize $\alpha_{rf^{l+}}$ and α_{rf^l} because all the variables involved in the above bounds are binary. The exact values of α_{rf^l} and $\alpha_{rf^{l+}}$ are completely determined from the bounds shown in Equations (14), (15), (16) and (17), all involving binary variables.

$$\alpha_{rf^{l+}} \leq \left(\sum_i \mathbb{I}^{l+}_{j^{XPU_r i}} + \mathbb{I}^{XPU,l}_r \right) /2 \tag{14}$$

$$\left(\sum_i \mathbb{I}^{l+}_{j^{XPU_r i}} + \mathbb{I}^{XPU,l}_r - 1 \right) /2 \leq \alpha_{rf^{l+}} \tag{15}$$

$$\alpha_{rf^l} \leq \left(\sum_i \mathbb{I}^{l}_{j^{XPU_r i}} + 1 - \mathbb{I}^{XPU,l}_r \right) /2 \tag{16}$$

$$\left(\sum_i \mathbb{I}^{l}_{j^{XPU_r i}} + 1 - \mathbb{I}^{XPU,l}_r - 1 \right) /2 \leq \alpha_{rf^l} \tag{17}$$

The additional constraints (18) and (19) linearize the constraint (12) involving binary variables.

$$\sum_i \mathbb{I}^{l+}_{ij^{XPU_r}} \leq \left(\sum_i \mathbb{I}^{l+}_{j^{XPU_r i}} + 1 - \mathbb{I}^{XPU,l}_r \right) /2 \tag{18}$$

$$\left(\sum_i \mathbb{I}^{l+}_{j^{XPU_r i}} + 1 - \mathbb{I}^{XPU,l}_r - 1 \right) /2 \leq \sum_i \mathbb{I}^{l+}_{ij^{XPU_r}} \tag{19}$$

Equations (20) to (22) capture the requirement of flow continuity in the intermediate nodes of the access/edge network.

$$\sum_i \mathbb{I}_{ij}^l = \sum_i \mathbb{I}_{ji}^l \tag{20}$$

$$\sum_i \mathbb{I}_{ij}^{l+} = \sum_i \mathbb{I}_{ji}^{l+} \tag{21}$$

$$\sum_i \mathbb{I}_{ijk}^l = \sum_i \mathbb{I}_{jik}^l \tag{22}$$

Equations (23), (24) and (25) capture the requirement of single path routing assumed in this paper.

$$\sum_j \mathbb{I}_{ij}^l \leq \mathbb{I}^l \quad \forall i\ node, \quad \forall\ fronthaul\ flow\ l \tag{23}$$

$$\sum_j \mathbb{I}_{ij}^{l+} \leq \mathbb{I}^l \quad \forall i\ node, \quad \forall\ fronthaul\ flow\ l \tag{24}$$

$$\sum_j \mathbb{I}_{ijk}^l \leq 1 - \mathbb{I}^l \quad \forall i\ node, \quad \forall\ backhaul\ flow\ k\ of\ source\ l \tag{25}$$

Finally, we also impose that some of the variables of the model are binary,

$$\begin{aligned}
\mathbb{I}^l \in \{0,1\} \quad & \forall f^l\ flow \\
\mathbb{I}_k^l \in \{0,1\} \quad & \forall b_k^l\ flow \\
\mathbb{I}_{ij}^l, \mathbb{I}_{ij}^{l+} \in \{0,1\} \quad & \forall (i,j)\ link,\ \forall f^l\ flow \\
\mathbb{I}_{ijk}^l \in \{0,1\} \quad & \forall (i,j)\ link,\ \forall b_k^l\ flow \\
\mathbb{I}_r^{XPU,l} \in \{0,1\} \quad & \forall r\ XPU,\ \forall f^l\ flow \\
\mathbb{I}_r^{XPU} \in \{0,1\} \quad & \forall r\ XPU
\end{aligned}$$

A major challenge in the general C-RAN deployment problem considered in this paper is to accommodate the stringent delay requirements of the fronthaul traffic (see Table 2). Consequently, the delay constraints should also be incorporated in the optimization. The following subsection presents the formulation of the (non-linear) delay constraints and the derivation of linear approximations to allow for employing linear programming solution tools.

3.2. Incorporation of Delay constraints

There are 3 delay components that every packet experiences between the completion of its arrival to a network node, say i, and that to the next node along its path, say node j. These delays will be attributed to link/port (i, j) and are determined by the transmission capacity (referred to as the transmission delay), the length (referred to as the propagation delay) and the queuing phenomena (referred to as the queuing delay) of link (i, j). Other sources of additional delay, such as that of processing time at the nodes or an XPU, will be considered to be relatively small and will be ignored. The consideration of the aforementioned 3 components will establish the impact of distances, capacities and traffic loads in a C-RAN environment, which is of utmost interest to the network operators. The transmission and propagation delays are easily derived, as the packet sizes, link distances and transmission capacities are assumed to be known. The challenge here is to derive the queuing delay in a way that is easily incorporated in the optimization formulation.

As the challenge in the C-RAN deployment is to ensure that the stringent delay constraints of the fronthaul flows is satisfied, a priority queuing scheme will be adopted giving non-preemptive priority to fronthaul packets over the backhaul ones. Although the sizes of the packets are considered to be fixed, we will adopt a queuing model with general service time, to keep the treatment more general. On the other hand, the arrival process will be considered to be Poisson, which is considered to be a reasonable model capturing the superposition of independent packet streams arriving over different input links to an outgoing link. Thus, we will consider an M/G/1 queuing model with 2 priority classes [30]. The packet arrival rates of priority n, λ_{ij}^n, can be expressed by Eq. (26).

$$\lambda_{ij}^1 = \sum_l f^l/p^l \cdot \mathbb{I}_{ij}^l, \ \lambda_{ij}^2 = \sum_l f^{l+}/p^l \cdot \mathbb{I}_{ij}^{l+} + \sum_{k,l} b_k^l/p_k^l \cdot \mathbb{I}_{ijk}^l \tag{26}$$

Let $\rho_{ij}^n = \lambda_{ij}^n/\mu_{ij}$ denote the traffic intensity at link (i, j) due to the incoming flows of priority class n. The classical queuing results provide for the mean queuing delay of packets of priority n, denoted by W_{ij}^n, described in Equations (27) and (28), where R_{ij} describes the mean remaining time till the completion of the transmission of the packet being transmitted upon a packet's arrival to node i; notice that since the packet size and link capacities are fixed, the second moment of the service time in (28) is equal to and has

been replaced by $1/\mu_{ij}^2$.

$$W_{ij}^n = \frac{R_{ij}}{(1-\rho_{ij}^1-\ldots-\rho_{ij}^n)(1-\rho_{ij}^1-\ldots-\rho_{ij}^{n-1})} \tag{27}$$

$$R_{ij} = \frac{\sum_n \lambda_{ij}^n/\mu_{ij}^2}{2} \tag{28}$$

Considering the packet transmission, queuing and propagation delay components over all links traversed by the flow (given by Equation (27)), the delay of a fronthaul packet in reaching its XPU is derived and given by Equation (29). This delay is subject to the most stringent constraint, as shown in Table 1. Considering the corresponding delay components similarly, the delay of a packet generated by a fronthaul source in reaching its destination is given by Equation (30), considering also its path (as a backhaul packet) from its XPU to its destination. Similarly, the delay experienced by a packet generated by a backhaul source is derived and given by Equation (31). Notice that a fronthaul packet has priority $n = 1$ while a backhaul packet has priority $n = 2$.

$$d^{l-} = \sum_{i,j} \frac{\mathbb{I}_{ij}^l}{\mu_{ij}} + \sum_{i,j} W_{ij}^1 \cdot \mathbb{I}_{ij}^l + \sum_{i,j} \frac{L_{ij}}{vl} \cdot \mathbb{I}_{ij}^l \tag{29}$$

$$d^{l} = d^{l-} + \sum_{i,j} \frac{\mathbb{I}_{ij}^{l+}}{\mu_{ij}} + \sum_{i,j} W_{ij}^2 \cdot \mathbb{I}_{ij}^{l+} + \sum_{i,j} \frac{L_{ij}}{vl} \cdot \mathbb{I}_{ij}^{l+} \tag{30}$$

$$d_k^l = \sum_{i,j} \frac{\mathbb{I}_{ljk}^l}{\mu_{ij}} + \sum_{i,j} W_{ij}^2 \cdot \mathbb{I}_{ijk}^l + \sum_{i,j} \frac{L_{ij}}{vl} \cdot \mathbb{I}_{ijk}^l \tag{31}$$

Notice that the delay expressions above include the non-linear functions W_{ij}^n (with respect to ρ_{ij} or λ_{ij}) which would not allow for the incorporation of linear programming tools for the solution of our optimization problem, even if most of the variables involved are binary. To address this problem, a linear approximation based on Taylor expansion along with an iterative procedure are adopted and are described next.

By considering the first terms of a Taylor expansion of W_{ij}^n around some point $(\rho_{ij}^{1,0}, \rho_{ij}^{2,0})$ we get the approximation $\tilde{W}_{ij}^n$ shown in Equation (32)

$$\tilde{W}_{ij}^{n} = \frac{1}{2\mu_{ij}} \cdot \left\{ a_{n0} + a_{n1} \cdot (\rho_{ij}^{1} - \rho_{ij}^{1,0}) + a_{n2} \cdot (\rho_{ij}^{2} - \rho_{ij}^{2,0}) \right\} \tag{32}$$

By substituting W_{ij}^{n} by $\tilde{W}_{ij}^{n}$ in Equations (29), (30) and (31) we end up with some products of variables. Since one of them is bounded (ρ_{ij}^{n}), and the other one is binary ($\mathbb{I}_{ij}^{l}$ or $\mathbb{I}_{ij}^{l+}$ or $\mathbb{I}_{ijk}^{l}$) we can linearize such products by introducing some additional variables, as shown for the case of the product in Equation (29) next.

$$y_{ij}^{l,n} = \mathbb{I}_{ij}^{l} \cdot \rho_{ij}^{n} \tag{33}$$

$$y_{ij}^{l,n} \leq \mathbb{I}_{ij}^{l} \tag{34}$$

$$y_{ij}^{l,n} \leq \rho_{ij}^{n} = \frac{\lambda_{ij}^{n}}{\mu_{ij}} \tag{35}$$

$$\mathbb{I}_{ij}^{l} + \rho_{ij}^{n} - 1 \leq y_{ij}^{l,n} \tag{36}$$

Finally, the following constraints are imposed on the delay of the fronthaul packets in reaching their XPU and their destination and the backhaul packets in reaching their destination.

$$d^{l-} \leq D^{l-} \cdot \mathbb{I}^{l} \tag{37}$$

$$d^{l} \leq D^{l} \cdot \mathbb{I}^{l} \tag{38}$$

$$d_{k}^{l} \leq D_{k}^{l} \cdot \left(1 - \mathbb{I}^{l}\right) \tag{39}$$

Since the linearized formula for the queuing delays shown in Equation (32) requires some (arbitrary) initial input for the class 1 and 2 traffic, some discussion on the impact of the particular approximation on the accuracy of the solution derived through the optimization framework is in order. A (first) solution to the optimization problem is obtained by considering an arbitrary initial value for the loads $(\rho_{ij}^{1,0}, \rho_{ij}^{2,0})$ in Equation (32). This solution determines also the loads and delays associated with all links. In the sequel, these loads are used for the calculation of the link delays based on the exact formula in Equation (27) and the result is compared with that returned by the solution to the optimization problem. If the deviation exceeds some threshold, then the new loads are considered as the initial values in Equation (32) and a new solution to the optimization problem is obtained yielding new loads and delays. The procedure continues until the aforementioned delay deviation is

below some accuracy threshold and the solution regarding the determined DUs and XPUs remains unchanged. A specific application of this approach is reported in Section 4.1.

3.3. Heuristic solution for the optimization problem

As it is proven in Appendix A the optimization problem considered in this paper is NP-complete. As a result, the computational complexity would be very high when large scale environments are considered. For such environments, an efficient heuristic of low computational complexity is proposed for solving the optimization problem prescribed in (1). The efficiency of the heuristic, which may yield the optimal or a suboptimal solution, is assessed in Section 4. The heuristic algorithm is described in detail in Algorithm 1 and it is outlined next.

```
All sources ← DU
DUsNotUsed ← DUs
while (UsedXPUs < MaxXPUs)&&(maxDUhit >
 1)&&(DUsNotUsed > NumberSources) do
    maxDUhit ← 0
    forall r ∈ XPUPlacement do
        maxDUXPU ← 0
        forall l ∈ DUsNotUsed do
            Path1_{f^l} ← ShortestPath(DU_l, XPU_r)
            while (Capacity(link) + f^l > MaxCapacity(link), link ∈
             Path1) and (Not All Links Removed) do
                Remove links that cannot transport f^l
                Path1_{f^l} ← ShortestPath(DU_{f^l}, XPU_r)
            end
            Path2_{f^l} ← ShortestPath(XPU_r, Destination)
            while (Capacity(link) + f^{l+} >
             MaxCapacity(link), link ∈ Path2) and
             (Not All Links Removed) do
                Remove links that cannot transport f^l
                Path2_{f^l} ← ShortestPath(XPU_r, Destination)
            end
            Recompute delays for flows already routed
            if Recomputed delays satisfy their maximum delay then
                Keep the paths and the DUs that are placed for the
                 current XPU
                maxDUXPU ← maxDUXPU + 1
            end
        end
        if maxDUXPU > maxDUXPUsaved then
            maxDUXPUsaved ← maxDUXPU
            Save the information for all the DUs that uses this XPU
        end
    end
    if maxDUXPUsaved > 1 then
        maxDUXPUit ← maxDUXPUsaved
        Save the information for all the DUs that uses this XPU
        Update DUsNotUsed removing the ones that uses the
         selected XPU
    end
end
```

```
flag ← 1
while flag == 1 do
    flag ← 0
    forall l ∈ DUsNotUsed do
        forall k ∈ BackhaulFlowsOfSource(l) do
            Path_{b_k^l} ← ShortestPath(source_l, Destination)
            while (Capacity(link) + b_k^l > MaxCapacity(link), link ∈
             Path) and (Not All Links Removed) do
                Remove links that cannot transport b_k^l
                Path_{b_k^l} ← ShortestPath(source_l, Destination)
            end
            Recompute delays for flows already routed
            if Recomputed delays satisfy their maximum delay then
                Keep the path of the new backhaul flow and update
                 the loads in the links
            end
            else
                flag ← 1
                Remove one DU from the XPU that accommodates
                 more DUs
                if The XPU selected contains only 2 DUs then
                    Remove the two DUs
                end
                Add the selected DUs to DUsNotUsed
                Update all the information saved for those DUs
            end
        end
    end
end
```

Algorithm 1: Heuristic 1

The algorithm aims at determining the best placements for the XPUs (supporting 2 or more DUs) while trying to accommodate as many DUs as possible. The algorithm starts by trying to accommodate the largest possible number of DUs that can be supported by one only XPU and determine the (best) placement of that one XPU. To accomplish this, the algorithm starts assuming that all the sources are DUs and the algorithm computes the paths and the associated loads/delays from the sources to each candidate XPU placement. The placement determined and the supported DUs are kept as the baseline for the next round of the algorithm. In the next round, the placement of one XPU that can accommodate the largest number of the remaining DUs

is determined. Following this, the new loads and delays are recalculated and the latest solution is accepted only as long as previous solutions are not invalidated; that is, the delay requirements of the flows whose paths were determined previously are not violated due to the new loads of the paths determined by the latest round. These rounds are repeated until the sources are exhausted or no more XPUs can be placed without invalidating previous placements.

At this point the XPUs and the supported DUs have been determined, including the paths from the DUs to the supporting XPU (fronthaul flow) and the path from the XPU to the destination (backhaul flow). The backhaul flows from the remaining sources (which are eNBs) are then routed to their destination; notice that the delay constraints are not as stringent for backhaul flows and that their loads are substantially less than that of the DU sources (fronthaul). Shortest path routing is considered for these backhaul flows, taking into account the remaining capacity of the links after having accommodated the flows of the DU sources. If the delay requirement of previously routed flows is not violated due to the shortest path routing of a backhaul flow, the determined path is accepted for the current flow. If the delay requirement of a previously routed flow is violated, the responsible links are "removed" (i.e., cannot be part of the route for the current flow) and the shortest path algorithm is reapplied until a path is found.

If no path is found for at least one (eNB) flow, we consider the XPU that accommodates the largest number of DUs and we switch one of those DUs to an eNB. The flow of that DU is removed, as well as the flows of all the eNBs routed before. If that XPU accommodates only two DUs, the XPU is removed and both DUs are removed since an XPU cannot support only one DU. The procedure for routing the eNB flows is then started again and is repeated until all such flows are routed.

At the end of this heuristic algorithm we obtain the largest possible number of DUs that can be accommodated, the number and placement of the supporting XPUs and the routes for all flows. This solution (regarding the number of DUs and XPUs) will be compared for some network topologies and scenarios against that returned by the optimal one obtained with a much higher computational complexity.

4. Validation/Application of the Approaches - Numerical Results

The optimization framework and the heuristic approach developed in this work are validated and evaluated by applying them over some topologies of practical interest in a Matlab environment.

In our research to find the best topology for the testing of our algorithms, we found that there is no single common topology used for the transport network of operators. Based on the available fiber deployment and geographical characterization, the operator may choose one topology or another. The only common rule followed is the use of aggregation of ring topologies, in which rings with lower bandwidths are aggregated in larger rings with higher capacity. Hence, following this spirit, we have selected a ring topology that is computationally feasible to validate the heuristic results compared with the linearized problem results. Moreover, to test the algorithm proposed we have selected a synthetic topology that follows the same principles given by the operators and the works [31] and [32], and can be characterized by a set of parameters which can be modified to assemble an operator deployment. Thus, the topology used for the experiments can be perfectly an example of a real operator deployment or can be parametrized to be similar to a real one.

4.1. Small-Scale Topology

First, a relatively small scale environment is considered in order to derive results under the optimization framework in reasonable computational time. That is, to determine the maximum number of DUs that can be accommodated with the minimum number of necessary XPUs each of which supporting two or more DUs. The derived solution is compared against that obtained under the heuristic approach introduced in Section 3.3 to assess the potential effectiveness of the heuristic. The network topology considered consists of a ring of 7 nodes connected with 10Gbps (per direction) bi-directional links and each of these nodes is connected to 3 traffic sources via a 1Gbps access link to each one of them, as shown in Figure 2.

A quick back-of-the-envelop calculation easily reveals that for the capacities and topology shown in Figure 2, the maximum number of DUs is 21 (all of the sources can be DUs) and the minimum number of XPUs is 1 (supporting all 21 DUs). Due to the symmetry in the topology, this XPU may be placed in any of the 7 nodes of the ring. The XPU will receive 9 of the non-local fronthaul flows (of a rate of 0.9Gbps each) over the clock-wise 10Gbps

ring and the other 9 non-local fronthaul flows over the counter-clock-wise ring of 10Gbps; the 3 local DUs are supported by the XPU without creating fronthaul traffic over the ring. Without loss of generality it is assumed that the 21 backhaul flows exiting the XPU facility are forwarded to destinations outside the shown network topology.

The aforementioned back-of-the-envelop result is just a way of explaining the results obtained by simulation and are the same as the ones obtained by solving the optimization problem and by applying the heuristic approach, validating both approaches. The back-of-the-envelop results are shown in Figure 3 and correspond to the value of $\rho = 1$ (denoting that the full capacity of the 10Gbps links is available, see discussion below). Notice that the number of XPUs is 1 and the total Air Bandwidth is equal to 4200 Mbps under both the optimization and the heuristic approaches. Assuming a cell Air Bandwidth of approximately 150 Mbps (LTE eNB node using 2x2 MIMO and 20MHz channel) and that a 33% Air Bandwidth gain is achieved when the eNB is replaced by an DU (which benefits from coordinated processing of its signals with those from at least one more cell [33]), then the total Air Bandwidth achieved by the 21 DUs is 21x200=4200Mbps, as shown in Figure 3 for $\rho = 1$.

As an iterative approach is needed for obtaining the optimization solution due to the queuing delay approximation (see discussion at the end of Section 3.2), the following may be reported for the solution obtained under the aforementioned experiment. The initial value for the loads are set to $(\rho_{ij}^{1,0}, \rho_{ij}^{2,0}) = (0.25, 0.25))$. Then, the loads in all links are determined. Their average values over all ρ_{ij} tuples were equal to $(\rho_{ij}^{1,1}, \rho_{ij}^{2,1}) = (0.2280, 0.0101)$ and the maximum value for $\rho_{ij}^{1,1}$ appeared in the tuple of $(\rho_{ij}^{1,1}, \rho_{ij}^{2,1}) = (0.9900, 0.0750)$; notice the lower values of load for the backhaul traffic (class 2) as this traffic imposes a lighter load and the solution determines that all the sources become DUs (generating fronthaul traffic till their CUs). With the new load values we iterate one more time and the final solution is reached and remains there after unchanged.

In order to test the performance of the developed approaches further, we expand the scenario considered in Figure 2 by considering that only ρ, $0.5 \leq \rho \leq 1$, of the ring capacity is available. As observed in Figure 3, the results obtained under both approaches coincide, demonstrating again the effectiveness of the heuristic approach. It may be noted that all 21 sources can be DUs, as the achieved Air Bandwidth remains equal to 4200Mbps, for

$0.5 \leq \rho \leq 1$. This fact, indicates that the same level of DU aggregation results are obtained by the application of the heuristic approach and the optimization framework. On the other hand the number of XPUs (i.e. XPU locations) required increases from 1 to 3, as the ring capacity decreases and the resulting fronthaul traffic cannot be forwarded to a single node any more. Note that although the numbers of XPUs and Air Bandwidth are the same for both approaches, some result details such as the location of the XPUs differ in both solutions. In addition, the resulting graphs in Figure 3 completely overlap due to the designed topology, which allows the deployment of all base stations as DUs.

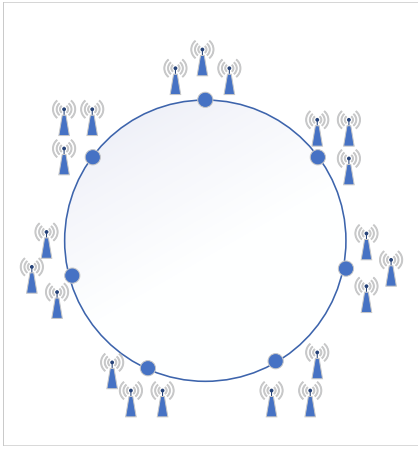


Figure 2: Small Scale Validation Environment

4.2. Large-Scale Topology / Practical Crosshaul Transport Network

In this subsection a large-scale network topology - that is likely to encounter in real environments - and some scenarios of potential interest to operators are considered. Due to the high computational complexity of the optimization framework, results are obtained by employing the heuristic approach of Section 3.3. These results turn out to provide for efficient deployment of C-RAN (that is, improved placement of the XPUs and accommodation of a large number of DUs). To this end, the Crosshaul transport network depicted in Figure 4 is considered that represents a real production transport network deployed in north Italy. This network has been provided by operator involved in the 5G-Crosshaul project [34]. It is based on a number of optical rings where the base stations are connected to. Each blue point in the rings

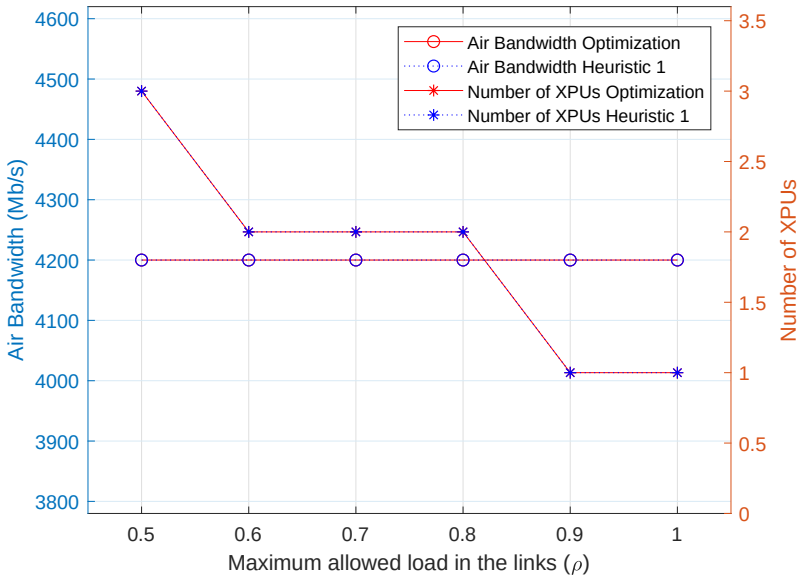


Figure 3: Optimization vs Heuristic 1 comparison

of Figure 4 corresponds to an Edge data center (potential host of an XPU facility). The length of each ring varies depending on the geographical area, ranging from 3Km to 100Km. This is the reference topology considered in this subsection.

Based on the scenario depicted in Figure 4, we have generated synthetic Ring-Tree based topologies as shown in Figure 5. Their configuration parameters (number of base stations, possible location for Edge data centers, radius of the links, etc.) are generated randomly. The generation process begins by forming hexagonal cells that form groups of size $A1$. Each of these hexagonal cells are supported by either a complete eNB node or by a DU.

Each of these cells is connected via a 1 Gbps link to one of the $A2$ nodes that reside on a ring of capacity of 10Gbps, which node is common to all the cells belonging to the same $A1$ group. $A3$ of those rings of $A2$ nodes are connected via in a ring of capacity of 40 Gbps. Finally, CR of those rings of $A3$ nodes are connected via a single ring CR (of capacity of 100 Gbps). Any of these CR nodes of the central ring would be considered to be the exit to the Internet where the destination of any flow generated within this topology would reside. Finally, any of the nodes residing in any

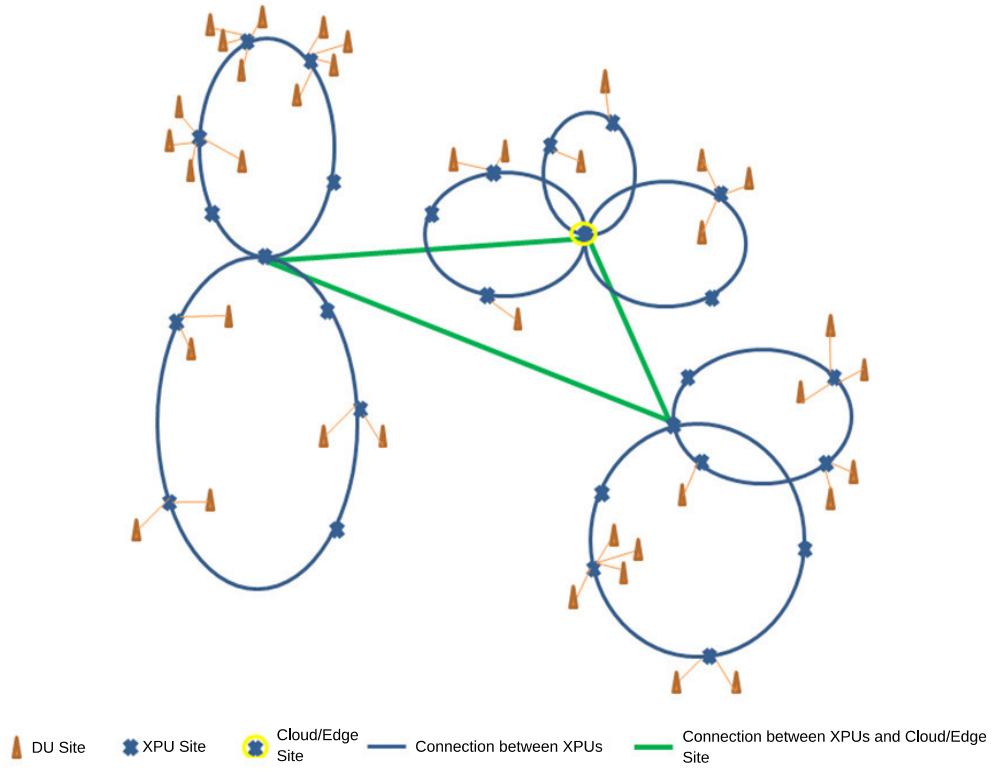


Figure 4: Reference topology

of the rings is a potential host of an XPU. For the rest of the section we will assume a topology with 339 nodes in total ($CR = 3$, $A3 = 5$, $A2 = 4$, $A1 = 6$). The objective of this section is to evaluate how a large scale operator network can be optimized based on our approach. In order to do so, we will stress the network based on 2 scaling parameters: i) the maximum end to end propagation delay in the network and ii) the maximum allowed capacity used in the links of the network, ρ, as we also did in the small scale environment case. With the first of these parameters we control the diameter of the network. This way, we can possibly have all XPUs placed in the central ring if the propagation delay is below the stringent delay constraint and the available capacities permit it; With the second parameter we control the available capacities in the network, which would also affect the placement of XPUs, depending on the induced queuing delays.

In the first of the experiments of this section, we derive and present results

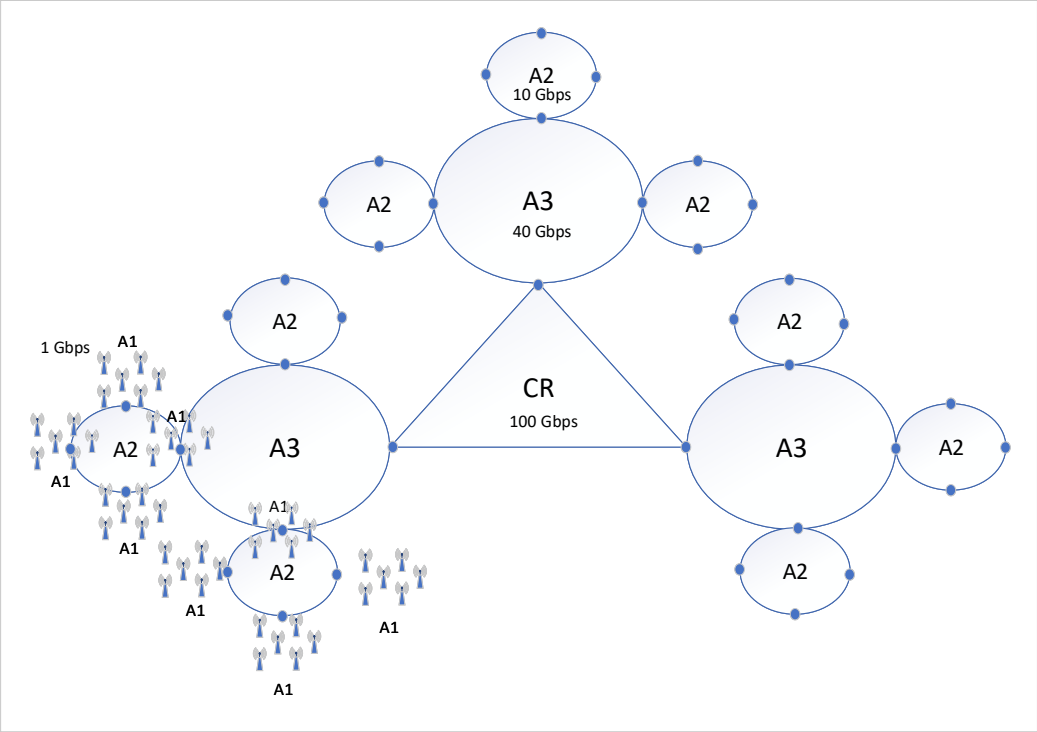


Figure 5: Synthetic Ring-Tree based topology

by applying Heuristic 1 (see Section 3.3) to the large scale network described above. Heuristic 1, determines the deployment mix of eNBs and DUs (and their placement) aiming at maximizing the Air Bandwidth (or number of DUs deployed), while minimizing the number of deployed XPUs. Figure 6 presents the results of Heuristic 1 for the Air Bandwidth (Figure 6(a)) and the Number of XPUs required (Figure 6(b)). Considering the Number of XPUs, Figure 6(b) shows how the number of XPUs deployed increases with the maximum propagation delay. The main reason for this behavior is that due to the delay increase, the aggregation of the flows of a high number of DUs in the higher aggregation rings (A3 and CR, in Figure 5) is not possible, requiring more XPUs and distributing them over the lower aggregation rings (A2 in Figure 5) to meet the fronthaul delay constraints. To illustrate this, consider the curve for $\rho = 1$ and compare the result corresponding to a propagation delay of $250\mu s$ with that of $1ms$. For the case of $1ms$, Heuristic 1 results in 12 XPUs: 3 XPUs placed in rings A3 and 9 in A2. For the case of $250\mu s$, Heuristic 1 places a total of 3 XPUs, placed 1 in the central ring

and 2 in the A3 rings. As explained earlier, the reason for this difference is that when delay constraints are met, the best solution is to aggregate in the higher aggregation rings.

Following a similar line of reasoning, when the maximum capacity of the links is reduced, from $\rho = 1$ to $\rho = 0.25$, the fronthaul traffic cannot be pushed deeper into the network, due to the saturation of the links in the aggregation rings. For this reason, the number of XPUs required increases while ρ decreases. To illustrate this, consider the result under a propagation delay of $500\mu s$ for $\rho = 0.25$ and $\rho = 1$. For the case of $\rho = 1$, the total number of XPUs is 6, placing 1 in a A3 ring and 5 of them in the A2 rings. For the case of $\rho = 0.25$, Heuristic 1 results in 27 XPUs, placing 1 in the central ring, 5 in A3 rings and 21 in A2 rings. As explained, the lower the bandwidth available (lower ρ), the more the XPUs required and the less the aggregation.

Figure 6(a) shows the resulting total Air Bandwidth of all DUs and eNBs, whose numbers are determined by Heuristic 1. As expected, Heuristic 1 achieves a higher level of aggregation under lower maximum propagation delay, determining a lower number of XPUs placed deeper in the network. For instance, under $500\mu s$ and $1ms$ maximum propagation delays, the number of XPUs increases (and they are pushed towards the edge of the network), compared with that under $250\mu s$. As a result, the number of DUs deployed will be decreased under low maximum propagation delay, since the fronthaul flows will share the bandwidth with backhaul flows for longer paths deeper into the network and the total Air Bandwidth will decrease, for all values of ρ. In addition, as expected, as ρ decreases, the resulting Air Bandwidth decreases accordingly. Note that this seems a different behavior from the one in Figure 3 where the air bandwidth does not decrease, it remains the same (4200 Mbps) for every value of ρ. It is not a different behavior, but in Figure 3 for the values of ρ selected the transport capacity still allows that all the sources are DUs, but due to the lack of transport bandwidth (lower ρ means lower link capacity) when ρ decreases the number of XPUs required to maintain the same air bandwidth has to increase. Here, in Figure 6(a) the lack of bandwidth in the links affects to the number of sources that can be DUs also and this is the reason it modifies the Air Bandwidth.

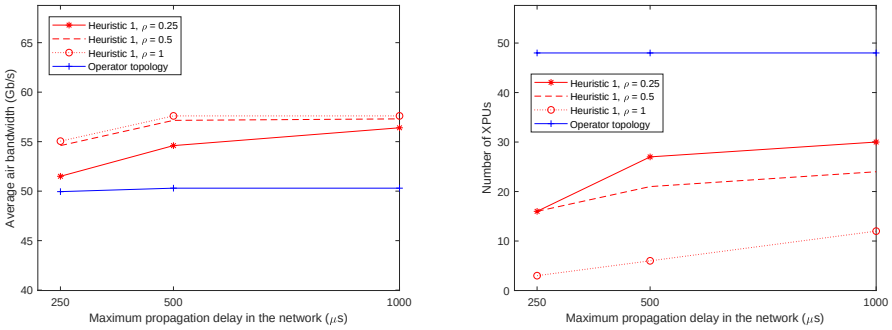
Finally, in Figure 6, we also provide a base-line to compare to. The line corresponding to the Operator topology represents the results that will be obtained by an operator deploying the same networks as used for Heuristic 1, but considering that the operator deploys just 1 XPU in the first point

aggregating the DUs (point corresponding to A1 in Figure 5), resulting in a higher number of XPUs (see Figure 6(b) and 6(a)) compared to the case under Heuristic 1. In addition, since the operator does not try to aggregate DUs, the pooling gain and Air Bandwidth gains based of cooperative signal processing cannot be obtained and the total Air Bandwidth is lower.

The proposed Heuristic 1 tries to optimize the network both in terms of Air Bandwidth (by choosing if a RAN element must be deployed as a eNB or a DU) and reduced number of XPUs. This is possible only if the RAN has not been already deployed or if the deployed RAN elements can be flexibly configured as eNBs or DUs. Since this is not always possible, as part of this work we have also developed a modification of Heuristic 1, called Heuristic 2 (see Appendix Appendix B), which takes as input a given topology with fixed RAN elements (i.e., whether they are eNB or DUs and their positions) and computes the minimum number of required XPUs. Results for Heuristic 2 are presented in Figure 7. As in Figure 6, we derive and present the Number of XPUs deployed in Figure 7(b) and the achievable total Air Bandwidth in Figure 7(a), for different values of the maximum propagation delay and different values of ρ. In order to build the simulated topologies we use the same ones as in Figure 6, but considering a probability of choosing eNB or DU $p_{DU} = 0,5$, resulting in an average of 144 DUs. The results in Figure 7(b) show that the Number of XPUs can be significantly reduced by applying the solution obtained by Heuristic 2, compared with the generic Operator deployment. Notice also a similar trend and for the same reasons as for Heuristic 1: the number of required XPUs increases with the maximum propagation delay and ρ. Regarding the Air Bandwidth, since all RAN elements are fixed, the bandwidth obtained by Heuristic 2 is similar to the Operator deployment, with a small gain due to the higher aggregation of DUs achieved. This small gain is already obtained with the lower value of ρ, thus the only value that changes when we increase the ρ is the number of XPUs required.

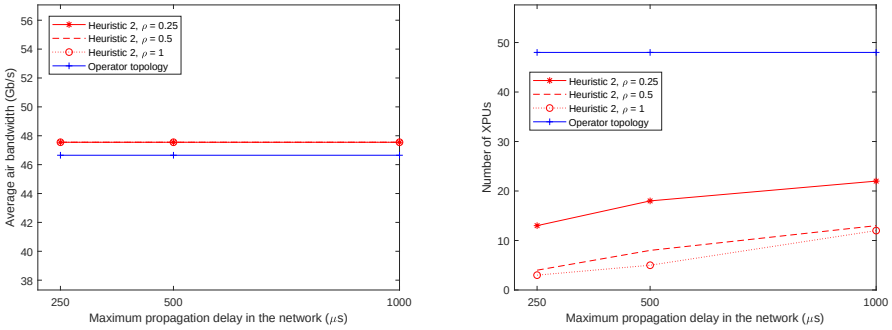
5. Conclusion

This paper has developed a framework for the joint optimization of an integrated networking and edge/cloud environment supporting two diverse classes of flows (fronthaul/backhaul) under path and delay constraints. This framework is directly applicable to the optimal design or dynamic management of a mixed Radio Access Network (RAN) and Cloud/Centralized-RAN



(a) Air Bandwidth (b) Number of XPUs required

Figure 6: Comparison of Heuristic 1 and a generic Operator deployment



(a) Air bandwidth (b) Number of XPUs required

Figure 7: Comparison of Heuristic 2 and a generic Operator deployment

(C-RAN) environments, foreseen on the road to 5G networking. These mixed environments emerge as operators attempt to maximize their adoption of the C-RAN technology in the most effective way, subject to the constraints imposed by the available supporting infrastructure. Or, such environments may emerge in a more dynamic (operation-level) case, where operators may switch on/off Distributed Units (DUs), such as Remote Radio Heads, or aggregate them in a lower number of Central Units (CUs), such as Base Band Units, according to the demand to reduce OPEX, necessitating the re-optimization of the resulting mixed RAN / C-RAN environment. The 5G networks incorporating a mixed RAN and C-RAN environment (where some nodes are split while others are not), will face planning and deployment challenges, requiring

mechanisms to decide on the most appropriate RAN element to split and the placement of the supporting CUs in the edge/cloud. It is also important to highlight that the use of split RAN elements requires the transport of the generated fronthaul flows characterized by more stringent throughput and delay requirements (than the RAN-generated backhaul flows) all the way to their CUs.

This paper provides an optimization framework and computationally less intensive heuristics to tackle exactly the aforementioned problems. The main contributions of this work are: i) an optimization framework for joint routing and resource placement is developed, taking into account delay, capacity and path constraints, maximizing the degree of DU deployment while minimizing the supporting CUs, ii) an efficient heuristic approach for solving the optimization problem in large scale environments, allowing the operator to derive solutions aiming at maximizing the Air Bandwidth (that is boosted by properly splitting a RAN element) while minimizing the number of XPUs (edge/cloud nodes hosting an array of CUs) by determining the placement of XPUs and the RAN elements that can be split into DUs and iii) a heuristic allowing the operator to compute the minimum number of XPUs and their placement for a given mixed RAN/C-RAN deployment. The approaches have been applied to both small scale and large scale/production level environments, demonstrating the effectiveness of the heuristics and the optimization approach and yielding potentially large gains in terms of reduced number of required Edge data-centers and increased Air Bandwidth.

Acknowledgment

This work was supported by EU H2020 project "5G-TRANSFORMER: 5G Mobile Transport Platform for Verticals" (grant no. 761536), EU H2020 5G-Coral Project (grant no. 761586) and by the Madrid Community through the BRADE project (P2013/ICE-2958 (BRADE-CM)).

References

[1] Cisco Visual Networking Index, Global mobile data traffic forecast update, 2016-2021, Cisco white paper.

[2] A. de la Oliva and J. A. Hernandez and D. Larrabeiti and A. Azcorra, An overview of the CPRI specification and its application to C-RAN-

based LTE scenarios, IEEE Communications Magazine 54 (2) (2016) 152–159. doi:10.1109/MCOM.2016.7402275.

[3] C. P. R. I. eCPRI Interface Specification, eCPRI specification v1.0 (August 2017).

[4] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, G. Fettweis, Fronthaul and backhaul requirements of flexibly centralized radio access networks, IEEE Wireless Communications 22 (5) (2015) 105–111. doi:10.1109/MWC.2015.7306544.

[5] 3GPP RAN3, TR 38.801 V14.0.0, Available at: `http://www.3gpp.org/ftp/Specs/archive/38\_series/38.801/` (Mar 2017).

[6] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna, A. Moored, 5g-crosshaul: An sdn/nfv integrated fronthaul/backhaul transport network architecture, IEEE Wireless Communications 24 (1) (2017) 38–45. doi:10.1109/MWC.2017.1600181WC.

[7] Mobile, China, C-RAN: the road towards green RAN, White Paper, ver 2.

[8] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, G. Wunder, 5G: A tutorial overview of standards, trials, challenges, deployment, and practice, IEEE Journal on Selected Areas in Communications 35 (6) (2017) 1201–1221. doi:10.1109/JSAC.2017.2692307.

[9] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, L. Dittmann, Cloud RAN for mobile networks - a technology overview, IEEE Communications Surveys Tutorials 17 (1) (2015) 405–426. doi:10.1109/COMST.2014.2355255.

[10] M. Peng, Y. Sun, X. Li, Z. Mao, C. Wang, Recent advances in cloud radio access networks: System architectures, key techniques, and open issues, IEEE Communications Surveys Tutorials 18 (3) (2016) 2282–2308. doi:10.1109/COMST.2016.2548658.

[11] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, S. V. Krishnamurthy, FluidNet: A flexible cloud-based radio access network for small

cells, IEEE/ACM Transactions on Networking 24 (2) (2016) 915–928. doi:10.1109/TNET.2015.2419979.

[12] A. Checko, A. P. Avramova, M. S. Berger, H. L. Christiansen, Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings, Journal of Communications and Networks 18 (2) (2016) 162–172. doi:10.1109/JCN.2016.000025.

[13] D. S. Reeves, H. F. Salama, A Distributed Algorithm for Delay-Constrained Unicast Routing, INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE 1 (1997) 84–91.

[14] A. Orda and R. Rom, Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length.

[15] R. Widyono, The Design and Evaluation of Routing Algorithms for Real-time Channels.

[16] M. R. Kabat, M. K. Patel, C. R. Tripathy, An Efficient Algorithm for Delay Delay-variation Bounded Least Cost Multicast Routing, International Journal of Computer, Electrical, Automation, Control and Information Engineering 3 (3).

[17] J. M. Smith, T. van Woensel, Topological network design of general, finite, multi-server queueing networks, European Journal of Operational Research.

[18] M. Garetto, D. Towsley, Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic, ACM 2003.

[19] M. Olvera-Cravioto, J. Blanchet, P. Glynn, On the Transition from Heavy Traffic to Heavy Tails for the M/G/1 Queue: The regularly varying case, The Annals of Applied Probability 21 (2) (2011) 645–668.

[20] J. M. Smith, M/G/c/K blocking probability models and system performance, Performance Evaluation 52 (2003) 237–267.

[21] J. MacGregor Smith, Properties and performance modelling of finite buffer M/G/1/K networks, Computers & Operations Research 38 (2011) 740–754.

[22] A. Gowda, J. A. HernÃąndez, D. Larrabeiti, L. Kazovsky, Delay analysis of mixed fronthaul and backhaul traffic under strict priority queueing discipline in a 5g packet transport network, Transactions on Emerging Telecommunications Technologies 28 (6). doi:10.1002/ett.3168.
URL http:https://doi.org/10.1002/ett.3168

[23] s. Agarwal and f. Malandrino and C. F. Chiasserini and S. De, Joint VNF Placement and CPU Allocation in 5G, IEEE INFOCOM 2018.

[24] A. Chakrabartiy and C. Chekuriz and A. Guptax and A. Kumar, Approximation Algorithms for the Unsplittable Flow Problem.

[25] H. Cho and A. Wein, Unsplittable Flows, in: Final Project, MIT.

[26] A. Karandikar, Approximation Algorithms for Stochastic Unsplittable Flow Problems, in: Master Thesis, School of Computer Science Computer Science Department Carnegie Mellon University Pittsburgh, PA, 2015.

[27] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, M.-J. Tsai, Deploying chains of virtual network functions: On the relation between link and server usage, in: Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, IEEE, 2016, pp. 1–9.

[28] T. Wan, P. Ashwood-Smith, A performance study of cpri over ethernet with ieee 802.1 qbu and 802.1 qbv enhancements, in: Global Communications Conference (GLOBECOM), 2015 IEEE, IEEE, 2015, pp. 1–6.

[29] 3GPP RAN3, Small cell virtualization functional splits and use cases version SCF159.07.02 Release 7, Available at: http://www.3gpp.org/ftp/Specs/archive/38_series/38.801/ (Accessed 29 May 2018) (January 2016).

[30] D. P. Bertsekas, R. Gallager, in: Data Networks, 1987, p. 186.

[31] N. Bram, K. Mario, V. Sofie, C. Didier, P. Mario, How can a mobile service provider reduce costs with softwareâĂŘdefined networking?, International Journal of Network Management 26 (1) 56–72. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.1919, doi:10.1002/nem.1919.

URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.1919

[32] Ruiquan Jing and Jianjun Tang and Luis Miguel Contreras Murillo and Rui Tang and Qiuyou Wu and Jean-Michel Caia and Yuanbin Zhang, Consideration on 5G transport network reference architecture and bandwidth requirements (January 2018).

[33] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, P. Camarda, Downlink packet scheduling in lte cellular networks: Key design issues and a survey, IEEE Communications Surveys & Tutorials 15 (2) (2013) 678–700.

[34] 5G-Crosshaul, D1.2: Final 5G-Crosshaul system design and economic analysis, 2017.

[35] D. P. Bertsekas, in: Network Optimization: Continuous and Discrete models, 1998, p. 349.

Appendix A. NP-Completeness

The developed optimization framework suffers from an exponential explosion of variables with respect to network size and the number of flows. By relating it to the multi-commodity flow problem with integer constraints (known to be NP-complete), it is shown in this Appendix to be NP-complete.

A **multi-commodity flow problem** involves a collection of several networks whose flows must independently satisfy conservation of flow constraints, but are coupled through some other constraints or the cost function. Consider a directed graph $(\mathcal{N}, \mathcal{A})$, and a finite collection of flow vectors $x(m), m = 1, ..., M$, on that graph, where M is a given integer. Let $x(m)$ denote the flow vector of commodity m, and let $x = (x(1), ..., x(M))$ denote the collection of all commodity flow vectors. Each flow vector $x(m)$ must satisfy its own conservation of flow constraints $\forall i \in N,\ m = 1, ..., M$,

$$\sum_{\{j|(i,j)\in A\}} x_{ij}(m) - \sum_{\{j|(j,i)\in A\}} x_{ji}(m) = s_i(m) \tag{A.1}$$

where $s_i(m)$ are given supply scalars. Furthermore, the commodity flows must together satisfy $x = (x(1), ..., x(M)) \in X$, where X is a constraint set, which may impose additional restrictions on the various commodities. For

example, to force a commodity m to avoid some arc (i,j), the constraint $x_{ij}(m) = 0$ may be introduced. In this way, one can model situations where each commodity is restricted to use only a subgraph of the given graph.

The feasible set is

$$F = \{x \in X | x \; satisfies \; Equation \; (A.1)\},$$

and the cost function is of the form $f(x) = f(x(1), ..., x(M))$.

The general *convex multi-commodity flow problem* is defined as

$$\begin{aligned} &minimize \quad f(x) \\ &subject \; to \quad x \in F \end{aligned}$$

where it is assumed that F is convex and f is convex over F.

Note that x may be viewed as a flow vector in an expanded graph consisting of M (disconnected) copies of the original graph $(\mathcal{N}, \mathcal{A})$. With this interpretation, it is seen that the only coupling between the commodities comes through the cost function and the constraint $x \in X$.

The version of the multi-commodity problem that is most amenable to analysis and algorithmic solution is the convex separable multi-commodity flow problem. In this problem the set X has the form

$$X = \{x | x_{ij}(m) \in X_{ij}(m), \forall (i,j) \in A, m = 1, ..., M\} \tag{A.2}$$

where $X_{ij}(m)$ are intervals of the real line and the cost function has the form

$$f(x) = \sum_{(i,j) \in A} f_{ij}(y_{ij}) \tag{A.3}$$

where y_{ij} is the total flow of arc (i,j), $y_{ij} = \sum_{m=1}^{M} x_{ij}(m)$ and each $f_{ij} : \mathbb{R} \to \mathbb{R}$ is a convex function of y_{ij}. Note here that the cost function is not separable with respect to the commodity flows $x_{ij}(m)$, but only with respect to the total flows y_{ij}. There is also a constraint-separable version of the multi-commodity flow problem, where the constraint set X has the form of Equation (A.2) but the cost function f does not have the separable form of Equation (A.3).

In the separable multi-commodity flow problem, commodities are coupled only through the total arc flows y_{ij} that appear in the separable cost function. Another type of commodity coupling in multi-commodity problems arises

when the set X includes additional upper bounds on the total flows of the arcs:

$$X = \{x | x_{ij}(m) \in X_{ij}(m), y_{ij} \leq c_{ij}\},$$

for all $(i,j) \in A, m = 1, ..., M$, where $X_{ij}(m)$ are given intervals of the real line, and c_{ij} are given scalars representing arc "capacities". The convex separable version of the resulting problem is referred to as a convex separable multi-commodity flow problem with arc capacities. This problem may also be viewed as a special case of the convex network problem with side constraints, where the side constraints are the capacity constraints $y_{ij} \leq c_{ij}$. The described multi-commodity flow problem is NP-complete when integer binary constraints are imposed on the side constraints.

In the sequel, the optimization problem considered here will be reduced to the multi-commodity flow problem to establish its NP-completeness. To this end, we employ the definition of the problem we know is NP-complete, the definition we have just given from [35], and then we construct our problem from the definition of the multi-commodity flow problem. In fact, the variant of this problem we use is the un-splittable flow problem because our flows have to follow a single path; that is, each flow can leave a node only through one link and cannot be split to follow several links or paths. The definition is the same but adding the constraint that flows cannot be split.

On one hand, we consider three types of flows (a fronthaul flow before reaching a CU (1), after leaving a CU (2) and backhaul flow (3)), and we consider one source that mixes all our sources, one destination that mixes all our destinations and one XPU that mixes all the XPUs, so the collection of all commodity flow vectors will be $x = (x(1), x(2), X(3))$ $(x(1) = x(2))$, then each vector $\forall i \in N,\ m = 1, 2, 3$

$$\sum_{\{j|(i,j)\in A\}} x_{ij}(m) - \sum_{\{j|(j,i)\in A\}} x_{ji}(m) = s_i(m) \tag{A.4}$$

$$s_i(1) = \begin{cases} 0 & if\ i\ is\ an\ intermediate\ node \\ x(1) & if\ i\ is\ a\ source, \\ -x(1) & if\ i\ is\ a\ XPU \end{cases}$$

$$s_i(2) = \begin{cases} 0 & if\ i\ is\ an\ intermediate\ node \\ x(2) & if\ i\ is\ a\ XPU, \\ -x(2) & if\ i\ is\ a\ destination \end{cases}$$

$$s_i(3) = \begin{cases} 0 & if\ i\ is\ an\ intermediate\ node\ or\ XPU \\ x(3) & if\ i\ is\ a\ source, \\ -x(3) & if\ i\ is\ a\ destination \end{cases}$$

In addition, we need to add a constraint in the set of constraints X to prevent the backhaul flows from entering XPUs,

$$\sum_{\{j|(i,j)\in A\}} x_{ij}(m) = 0, \quad \forall i\ a\ XPU,\ m = 3 \tag{A.5}$$

On the other hand, we need to add the constraint for the capacities of the links as in the multi-commodity flow problem with arc capacities. The constraint will be introduced as follows: y_{ij} is the total flow of arc (i, j) $y_{ij} = \sum_{m=1}^{3} x_{ij}(m)$ and the side constraints are the capacity constraints $y_{ij} \leq c_{ij}$, where c_{ij} is the capacity of the link (i, j). Also, the rest of the constraints in our framework will be introduced in the set X as constraints of each type of commodity.

Furthermore, as with the multi-commodity flow problem, the commodity flows must together satisfy $x = (x(1), x(2), x(3)) \in X$, where X is a constraint set, which may impose additional restrictions on the various commodities beyond those in Equation (A.5).

The feasible set is

$$F = \{x \in X | x\ satisfies\ (A.4)\ and\ the\ capacities\ of\ links\}$$

and the cost function is of the form $f(x) = f(x(1), x(3))$.

$$f(x(m)) = \begin{cases} -1 & if\ the\ source\ is\ a\ DU \\ 1 & if\ the\ node\ is\ an\ XPU \\ 0 & otherwise \end{cases}$$

And the general *multi-commodity flow problem* becomes

$$\begin{aligned} &minimize && f(x) \\ &subject\ to && x \in F \end{aligned}$$

where we assume that F is convex and f is convex over F.

Concluding, since we can reduce the multi-commodity flow problem with side constraints with integer values to our problem and the first one is NP-complete, our problem is also NP-complete.

Appendix B. Heuristic algorithm for fixed RAN elements (Heuristic 2)

$DUsNotUsed \leftarrow DUs$
while $(UsedXPUs < MaxXPUs)\&\&(maxDUit > 1)\&\&(DUsUsed < NumberDUs)$ **do**
  $maxDUit \leftarrow 0$
  forall $r \in XPUPlacement$ **do**
    $maxDUXPU \leftarrow 0$
    forall $l \in DUsNotUsed$ **do**
      $Path1_{f^l} \leftarrow ShortestPath(DU_l, XPU_r)$
      while *(*$Capacity(link) + f^l > MaxCapacity(link),\ link \in Path1$*) and (Not All Links Removed)* **do**
        Remove links that cannot transport f^l
        $Path1_{f^l} \leftarrow ShortestPath(DU_{f^l}, XPU_r)$
      end
      $Path2_{f^l} \leftarrow ShortestPath(XPU_r, Destination)$
      while *(*$Capacity(link) + f^{l+} > MaxCapacity(link),\ link \in Path2$*) and (Not All Links Removed)* **do**
        Remove links that cannot transport f^l
        $Path2_{f^l} \leftarrow ShortestPath(XPU_r, Destination)$
      end
      Recompute delays for flows already routed
      if *Recomputed delays satisfy their maximum delay* **then**
        Keep the paths and the DUs that are placed for the current XPU
        $maxDUXPU \leftarrow maxDUXPU + 1$
      end
    end
    if $maxDUXPU > maxDUXPUsaved$ **then**
      $maxDUXPUsaved \leftarrow maxDUXPU$
      Save the information for all the DUs that uses this XPU
    end
  end
  if $maxDUXPUsaved > 1$ **then**
    $maxDUXPUit \leftarrow maxDUXPUsaved$
    Save the information for all the DUs that uses this XPU
    Update $DUsNotUsed$ removing the ones that uses the selected XPU
  end
end

```
flag ← 1
while flag == 1 do
    flag ← 0
    forall l ∈ eNBs do
        forall k ∈ BackhaulFlowsOfeNB(l) do
            Path_{b_k^l} ← ShortestPath(source_l, Destination)
            while (Capacity(link) + b_k^l > MaxCapacity(link), link ∈
             Path) and (Not All Links Removed) do
                Remove links that cannot transport b_k^l
                Path_{b_k^l} ← ShortestPath(source_l, Destination)
            end
            Recompute delays for flows already routed
            if Recomputed delays satisfy their maximum delay then
                Keep the path of the new backhaul flow and update
                 the loads in the links
            end
            else
                flag ← 1
                Remove one DU from the XPU that accommodates
                 more DUs
                if The XPU selected contains only 2 DUs then
                    Remove the two DUs
                end
                Accommodate to other XPU
                Update all the information saved for those DUs
                Remove the information of the backhaul flows placed
                Look for another XPU to accommodate the DU
            end
        end
    end
end
```

Algorithm 2: Heuristic 2