Multi-Objective Function Splitting and Placement of Network Slices in 5G Mobile Networks

Multi-Objective Function Splitting and Placement of Network Slices in 5G Mobile Networks

Jalolliddin Yusupov*, Adlen Ksentini†, Guido Marchetto*, Riccardo Sisto*
* Politecnico di Torino
Email: {name.surname}@polito.it
† Eurecom
Email: adlen.ksentini@eurecom.fr

***Abstract*—5G networks are expected to support various applications with diverse requirements in terms of latency, data rates and traffic volume. Because of this, the selection of the appropriate 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, we explore two possible solutions. We propose a Mixed Integer Quadratically Constrained Programming (MIQCP) model for the efficient placement of Virtualized Network Function (VNF) chains in future 5G systems, with particular emphasis on different aspects of the functional split between the cloud platform and the radio access points. Then, we also express the placement problem as MaxSAT instance and provide formal assurance of policies by considering increasingly relevant scenarios where the radio access network (RAN) needs to support various slices. Hence, thorough analyses are performed and recommendation for split point between central cloud and distributed radio units are discussed in this paper.**

***Index Terms*—5G, functionality split, Virtual Network Embedding, uRLLC, eMBB.**

I. Introduction

The 5G ecosystem involves a number of vertical markets, such as automotive, smart grid, and the Internet of Things (IoT), and supports a number of use cases, with extreme diversity in service requirements with respect to throughput, latency, reliability, availability, as well as energy efficiency and cost efficiency. In order to satisfy these requirements, the concept of network slicing has been proposed as a means of sharing of a single network infrastructure between multiple network operators, where each operator provides specific way of handling the control and data (user) plane for delivering services to its users. Network slices are composed of a collection of network functions and specific radio access technology (RAT) settings, which are combined together for the specific use case. Moreover there is a definition of network slice categories, wherein each 5G use case may fall: eMBB (enhanced Mobile Broadband), uRLLC (Ultra Reliable Low Latency Communications), mMTC (massive Machine Type Communications). Network Slices are built on top of a NFV architecture, where a set of VNFs is instantiated on demand, and orchestrated to manage their lifecycle. The set of VNFs (may be) instantiated over a federated cloud, and interconnected with the RAT to build an end-to-end slice tailored to specific service. With the recent advance, the RAN is desegregated in specific network functions, which could be virtualized and run as VNF.

Indeed, unlike LTE, 5G New Radio disaggregates the RAN by introducing the CU, DU, and RU three-tiered (two-level Fronthaul structure) architecture, where the Distribution Unit (DU) hosts time-critical L1/L2 functions and aggregates a subset of Remote Radio Units (RRUs) (equivalent to BBU-baseband processing unit), and Centralized Unit (CU) remaining functions. This in turn, better facilitates RAN virtualization and provides more capability, flexibility and scalability compared to 4G RAN architecture[1]. According to [2], 3GPP RAN3 introduced 8 functional split options on the protocol stack to achieve a standard split for the CU/DU/RU architecture. In particular, the choice of functional split will determine the transport capacity requirement and associated latency specifications and performance. This will impact the network slicing as, for example, it can determine the placement of nodes and distance between them.

Focusing on uRLLC, eMBB and mMTC, in this paper we analyze how low-latency and high bandwidth requirements of these traffic classes are met by providing different split between CU and DU. The analysis is performed by finding the optimal placement for the service function chains based on optimization goals for different network slices. We present two approaches for finding the placement of the RAN functions and chaining them together taking into account the limited network resources and requirements of the functions. First we formulate the placement problem as MIQCP for the RAN requirements in terms of latency and throughput. Then we adopt our approach presented in [3] to provide formal verification and optimal placement of VNFs with propositional logic formulas in Conjunctive Normal Form.

The rest of the paper is structured as follows. We briefly discuss the related work in Section II. We present the problem formulation in Section III and IV. In Section V we evaluate our methodology by means of different use cases and present the results in Section VI. Section VII concludes the paper.

II. Concept, Definitions, and Related Work

In the 3GPP Evolved Packet System (EPS), network functions are grouped before running as independent functions, e.g., enhanced Node B (eNB), Serving Gateway (S-GW), or Mobility Management Entity (MME). This results in static

function assignments, where the function placement plan is already decided, preventing flexible network configurations or dynamic service deployments. This has led existing works in the literature towards flexible solutions to reduce the deployment cost, and accomplish (i) split of network elements into multiple basic network functions and (ii) optimally placing and chaining these basic network functions.

The functional split problem has attracted significant attention from the scientific community and works ([4], [5], [6], [7], [8], [9], [10]) show that a more flexible split is possible. For example, [5] argues that moving the RAN functionality towards cloud environments is more desirable, due to the lack of high speed optical links, especially in urban small cell environments. Whereas, [9] proposes a RAN configuration in terms of splitting the radio and baseband functionalities between CU and RU, and shows the impact of this flexibility on delivery of uRLLC, mMTC and eMBB applications. Another factor of varios functional splits is different hardware options on the implementation of RAN functions and it is addressed by [11].

One of the limitations of the previously discussed works is that they only focus on fully distributed or fully centralized baseband deployments. Instead, Cloud-RRH [12] proposes a concept of hierarchical infrastructure by placing an intermediate edge cloud close to mobile user. It allows to have a flexible functional split of the radio protocol that simplifies network management and enables resource pooling and coordination of radio resources.

While in the literature many works [8], [13], [14] address the problem of functional placement, selection of the optimal functional split and its placement still remains a challenging task, since a number of parameters have to be considered in order to make such a decision. In [15], which is the closest to our approach, authors address the placement problem in terms of functional split options between CU/DU/RU. Instead, we formulate the virtual network embedding (VNE) problem based on slice requirements of different traffic classes targeting different objectives and focusing on functional split of individual radio functions. Similarly, the impact of parameters on the functional split is addressed by [8]. The authors formalize and solve a dynamic VNE for 5G networks supporting different functional split options.

III. Placement Model and Problem Formulation

In this section, we list the required input considered in the placement algorithm, and then we present the MIQCP formulation of the function placement problem.

We model the substrate network, where RAN network function chains are placed, as a connected directed graph, $G = (V, E)$. The data rate available is $d(v, v')$ for every edge $(v, v') \in E$ and the network links are directed edges with latency $l(v, v')$. Upon the arrival of a service request, an orchestrator component of NFV must decide how to optimally allocate the different ordered sets of radio network functions onto the substrate network nodes. In these requests, depending on the user-plane (UP) and control-plane (CP), different order of functions are specified, which define flows between fixed start (e.g., user equipment) and end (e.g., IP services) points. Set $A_{pairs} \subseteq A \times A$ of pairs of start and end points belonging to different flows.

We model the placement optimization problem as an MIQCP with respect to number of used network nodes, latency and data rate. Capacity of network nodes and requirements of different network functions then characterize the input. The notation used in our derivation is summarized in Table I, where *lat* and *rem* are continuous variables.

TABLE I: Summary of key notations

Domain	Parameter	Description
$\forall v \in V$	$c(v)$	Substrate node computational resources in v
$\forall (v,\ v') \in E$	$l(v, v')$ $d(v, v')$	Latency of (v, v') Data rate capacity on (v, v')
$\forall (u, u') \in U_{pairs}$	$d_{req}(u, u')$	Data rate demand of (u, u')
$\forall (u, u') \in U_{pairs}$	$l(u, u')$	Latency between u and u'
$\forall u \in U$	$p(u)$	Substrate node demand of u
$\forall (a, a') \in A_{pairs}$	$paths(a, a')$ $l_{req}(a, a')$	Paths between a and a' Required latency between a and a'
$\forall u \in U, \forall v \in V$	$m_{u,v}$	u mapped to v
$\forall (v, v') \in E,$ $\forall x, x' \in V,$ $\forall (u, u') \in U_{pairs}$	$e_{v,v',x,y,u,u'}$	(v, v') belongs to path between x and y, where u and u' are mapped to
$\forall v \in V$	$used_v$	At least one request mapped to v
$\forall (v, v') \in E$	$rem_{v,v'}$	Remaining data rate on (v, v'), when services utilize the same link

1) Placement Constraints: By Formula $\forall u \in U : \sum_{v \in V} m_{u,v} = 1$, all virtual nodes must be mapped onto a single substrate node, iff. the request is to be embedded. If at least one function is mapped on a substrate node, we denote it as "used" with the constraint: $\forall u \in U, \forall v \in V : \quad m_{u,v} \leq used_v$

Resource requirements such as a required storage of all functions mapped to a node should be less than or equal to available resources in that node:

$$\forall v \in V : \quad \sum_{u \in U} m_{u,v} \cdot p(u) \leq c(v) \tag{1}$$

In addition, we must ensure that the number of allocated nodes on the substrate node is less than the allowed number of network functions.

2) Path Related Constraints:: If (u, u') pairs are mapped to the (x, y) nodes and an edge in the request belongs to a path between nodes v and v', then the path is created between those network nodes:

$$\forall (v,\ v') \in E, \forall x,\ y \in V, \forall (u,\ u') \in U_{pairs} : \\ e_{v,v',x,y,u,u'} \leq m_{u,x} \cdot m_{u',y} \tag{2}$$

Moreover, each functional split has different latency requirements for data transfers across the function locations; involves different amounts and types of resources (computing power, link capacities); and brings different cost savings and performance benefit. This is expressed by the following constraints:

$$\forall (v, v') \in E, x, y \in V, (u, u') \in paths(a, a') : \\ e_{v,v',x,y,u,u'} \cdot l(v, v') \leq l_{req}(v, v') \tag{3}$$

According to uRLLC requirements where the sum of latencies of all edges of a flow should be less than the maximum latency given for that flow is expressed as follows:

$$\forall (a,a') \in A_{pairs}: \sum_{\substack{(v,v')\in E, x,y\in V,\\ (u,u')\in paths(a,a')}} e_{v,v',x,y,u,u'} \cdot l(v,v') \leq l_{req}(a,a') \quad (4)$$

The total bandwidth consumed by all the requested source-destination traffic flows going through an edge should not exceed the bandwidth capacity of this edge:

$$\forall (v,v') \in E: \sum_{(u,u')\in U_{pairs}, \forall x,y\in V} e_{v,v',x,y,u,u'} \cdot d_{req}(u,u') \leq d(v,v') \quad (5)$$

Sum of latencies of network edges belonging to a path where u and u' are mapped gives the end-to-end latency of this path:

$$\forall (u,u') \in U_{pairs}: l(u,u') = \sum_{x,y\in V,(v,v')\in E} e_{v,v',x,y,u,u'} \cdot l(v,v') \quad (6)$$

Finally, the remaining data rate of an edge is calculated as,

$$\forall (v,v') \in E: rem_{v,v'} = d(v,v') - \sum_{\substack{(u,u')\in U_{pairs},\\ \forall x,y\in V}} e_{v,v',x,y,u,u'} \cdot d_{req}(u,u')$$

3) Objectives: Combination of different objectives can be targeted for different use case scenarios, and each of them can result in a different mapping of the network functions into the network graph. This section describes the multi-objective algorithm that aims to find the best candidate node for embedding each VNF of a chain. This algorithm consists of three steps:

Minimizing the number of utilized nodes in the network:

$$minimize \sum_{v\in V} cost(used_v) \quad (7)$$

This objective aims to minimize the cost of utilized DCs and applicable for all type of services. The cost helps businesses to rent the hardware or software from infrastructure providers, paying only for what they use. However, it might concentrate the placement of functions, which causes congestion in the network. In fact, centralizing RAN functions within a cloud infrastructure significantly improves cost but they can only be centralized so far as the latency budget is still met, plus other factors such as transport capability.

Maximizing the remaining data rate on network links:

$$maximize \sum_{(v,v')\in E, v\neq v'} rem_{v,v'} \quad (8)$$

In order to avoid the congestion in the network, we introduce this objective that maximizes the data rate on the links leaving more bandwidth for future requests and it is only applied for eMBB services.

Minimizing the latency of the created paths: As previously stated, 5G is supposed to provide users with unprecedented experience with ultra-low latency. However, there might be multiple paths available between the endpoints that are all compliant with low latency constraints. In these instances, selection of the path with the minimum latency is favorable and it can be expressed with the following objective:

$$minimize \sum_{(a,a')\in l_{req}} \Big(\sum_{P\in paths(a,a')} \Big(\sum_{(u,u')\in P} l(u,u') \Big) \Big) \quad (9)$$

IV. Formal verification

Mathematical programming presented earlier is a fundamental combinatorial optimization problem and it is restricted to take binary, integer, or real values. In this section, we also formulate the placement problem as Maximum Satisfiability (MaxSAT) model, which has a higher descriptive power than classical mixed-integer linear programming languages.

These "high-level" constraints allows us to provide formal assurance that the selected function chain correctly implements the reachability policies and that, at the same time, the latency requirements are met. The goal of verification is then to find a truth assignment for all the logical variables of the model that makes true all the logic formulas (clauses) in the network model, if one exists. This is the traditional form of the well-known Satisfiability (SAT) problem, where all the formulas are "hard" clauses and must be satisfied. This set of hard clauses represents in our case the network function forwarding behavior models, the reachability properties that we want to ensure, and the hard constraints we have on placement (e.g. we cannot exceed the resources available in each infrastructure node).

Problem Formalization. To formalize the placement problem, we introduce boolean variables y_i and x_{ij} that take true value when substrate node n_i^s is in use and when network function n_i^v is hosted on substrate node n_j^s, respectively. This last predicate is also denoted $n_i^v \uparrow n_j^s$. The mapping of a service request is then represented by two mapping functions: M_n, which maps network functions of the service request onto substrate nodes that meet their resource requirements as before, and M_e, which maps endpoints. M_n can be formally defined as follows. For all $n^v \in N^v$, $M_n(n^v) = n^s$, subject to $n^s \in N^s$, and $n^v \uparrow n^s$, and, for each j such that $n_j^s \in N^s$,

$$\left(\sum_{\forall i | n_i^v \uparrow n_j^s} storage(n_i^v) * x_{ij} \right) \leq storage(n_j^s) * y_j \quad (10)$$

where we are assuming the $true$ value of x_{ij} and y_j corresponds to 1, while their $false$ value corresponds to 0.

In Equation 10, we are assuming that network function from the same service request can share the same substrate node, which is common in NFV systems, e.g., in order to reduce latency.

Given this formalization, we build the set of clauses to represent M_n as follows. First of all we include, as hard

clauses, the inequalities in (10). In addition to these inequalities, we need to represent explicitly that M_n is a function, i.e. it maps each function onto exactly one node. For each i such that $n_i^v \in N^v$, this constraint is expressed by the following equation: $\sum_{\forall j | n_j^s \in N^s} x_{ij} = 1$.

Finally, for each j such that $n_j^s \in N^s$, in order to correctly bind variable y_j to variables x_{ij}, we add the implication $y_j \implies \bigvee_i x_{ij}$, i.e., when substrate node n_j is in use, there is at least one network function deployed on this node.

Routing tables. The network behavior of the virtual service is modeled by a set of formulas that represent the routing tables of each network function involved in the service request. These formulas express the next hops - next gateways to which packets must be forwarded along the path to their final destination. For each function v_i^v and its adjacent one-hop neighbor v_{adj}^v, we define a predicate $route(v_i^v, v_{adj}^v, l^s)$ which is true if the adjacent neighbor of v_i^v is v_{adj}^v and it is reached via link l^s.

The routing table of the endpoint device e_n^v in the chain is formulated as a set of soft clauses, with the opposite of the link latency as the weight. In this way, the MaxSAT solver will minimize the overall latency of the chosen path in the infrastructure. As the location of e_0^v is fixed in the substrate endpoint e_0^s, we generate the following soft constraint for each possible substrate node n_k^s onto which n_{adj}^v (adjacent neighbor function in the chain) can be allocated

$$Soft((route(e_0^v, n_{adj}^v, l_{0k}^s) \implies x_{adj\ k}), -latency(l_{0k}^s))$$

where the notation $Soft(c, w)$ specifies that clause c is a soft clause with weight w. In practice, the routing table of the endpoint device specifies to which substrate node k a packet is forwarded depending on the allocation of the next network function in the service chain. The soft clauses related to the other network function $n_i^v \in N^v$, with $i > 0$, are formulated similarly:

$$Soft((route(n_i^v, n_{adj}^v, l_{jk}^s) \implies x_{ij} \wedge x_{(adj)k}), -latency(l_{jk}^s))$$

i.e., if network function i forwards packets to the adjacent one adj in the service graph through link l_{jk}, then the corresponding boolean variables x_{ij} and $x_{(adj)k}$, which indicate the locations of the network functions, must be true. If two of them are allocated onto the same substrate node, i.e., $j = k$, we have $latency(l_{jk}^s) = 0$, and a soft clause with weight equal to zero is added to the set.

Configuration parameters of network functions allow us to model a fixed processing delay for each. This is represented by the $latency(n^v)$ function, which can be used in order to include the processing delay of the given network function when computing the overall end-to-end latency. If we have an upper bound on the overall end-to-end latency that must be guaranteed in the system, we can formulate it as an additional hard clause.

Optimization Objectives. As we noted before, RAN function placement is a multi-objective optimization problem. From a network infrastructure perspective, as many service requests as possible should be mapped onto the substrate network, making efficient use of the substrate network resources. However, the uRLLC environment usually requires minimization of link propagation delay between the endpoints too. Accordingly, the objective function of our formulation has two goals for this specific scenario: to minimize the number of substrate nodes in use and to minimize network latency.

The soft clauses involving the $route$ predicates cause the solver to minimize latency. In order to minimize the number of substrate nodes in use we add the following additional soft clause for each substrate node: $n_i^s \in N^s$: $Soft(\neg y_i, K)$ where K is a constant selected according to whether we want to give priority to latency minimization or to number of substrate nodes in use minimization. The MaxSAT solver attempts to assign $false$ values to the boolean variables y_i in order to minimize the penalty for falsified clauses in the current model, thus minimizing the number of nodes in use. Then, if we feed the set of formulas defined so far along with models in Section IV-A to the MaxSAT solver, it returns, if possible, a model that satisfies all hard clauses, including the ones about reachability (see Section IV-A), while minimizing latency and the number of nodes in use.

A. Verification approach

The modeling approach of the Verigraph tool presented in [16] is particularly interesting for our work as it is completely compatible with the z3Opt solver. In particular, by means of this approach we can statically analyze network configurations of the SG to check the satisfiability of network policies such as reachability or isolation.

The works presented in [16], [3] must be extended in order to be applied in our solution. First, we need to modify the presented packet forwarding model to make the approach more efficient. In particular, we can eliminate the notion of quantitative time that is currently used in Verigraph models in order to reduce the size of the problem and thus speedup the verification process in complex scenarios. Secondly, we simplify the forwarding behaviour of network functions in order to comply with RAN specific use cases. In the following we present the forwarding model of the network employed in this paper, as derived by the Verigraph approach.

The general forwarding behavior of a network can be expressed by means of the following set of conditions imposed on those two functions:

$$\begin{aligned} send(n_0, n_1, p_0) \implies & (n_0 \neq n_1 \wedge p_0.src \neq p_0.dest \wedge \\ & sport(p_0) \geq 0 \wedge sport(p_0) < MAX\_PORT \wedge \\ & dport(p_0) \geq 0 \wedge dport(p_0) < MAX\_PORT \wedge), \end{aligned} \tag{11a}$$

$$recv(n_0, n_1, p_0) \implies send(n_0, n_1, p_0), \quad \forall n_0, p_0 \tag{11b}$$

Formula 11a states that the source and destination nodes (n_0 and n_1) must be different, as well as the source and destination addresses in the packet ($p_0.src$ and $p_0.dest$). The source and destination ports must also be defined in a valid range of values. If a packet is received by a node (n_1), this implies that the packet was sent to this node. This is expressed by Formula 11b.

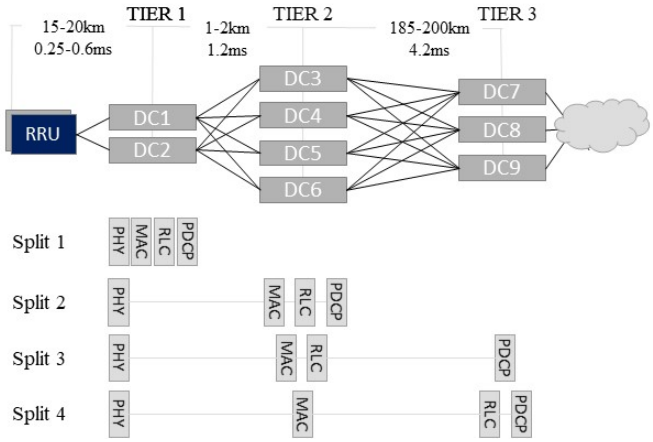

Fig. 1: 5G RAN substrate network topology

We can verify the reachability between the src and $dest$ nodes in presence of a set of radio functions thanks to the following formula:

$$\exists(n_0, p_0) \mid recv(n_0, dest, p_0) \wedge p_0.origin == src \quad (12)$$

Here we are modeling the case of a source node (src) that is sending a packet to a destination node ($dest$) of which we want to check the reachability. We must also impose that the destination node receives a new packet (p_0) from the last node (n_0): the received packet must have the source node as origin ($p_0.origin == src$).

In addition to the above clauses, it is necessary to impose formulas describing the specific behavior of involved network functions. In the Formula 13, we present a generic network function - radio, that forwards packets towards destination.

$$\begin{aligned}(send(radio, n_0, p_0)) \implies (\exists(n_1, p_1) | recv(n_1, radio, p_1) \wedge \\ p_1.src = p_0.src \wedge p_1.inner\_dest = p_0.inner\_dest \wedge \\ p_1.orig\_body = p_0.orig\_body \wedge p_1.inner\_src = p_0.inner\_src, \\ \forall(n_0, p_0)\end{aligned} \quad (13)$$

Conjunction of the placement constraints with the forwarding behavior of the network represents the overall model of the system. By feeding the solver with this input, we obtain the placement plan that ensures the network-wide properties are satisfied.

V. Use Cases

In this section, we introduce different representative use case scenarios of uRLLC and eMBB, which have very different resource requirements. Instead, we don't consider mMTC slices, which requires to support huge number of devices at lesser cost and enhanced coverage along with long battery life. Generally, network slicing involves the deployment of the entire mobile network functions and components to run on virtual platforms, i.e., VMs running on data centers. Figure 1 represents a realistic hierarchical 5G RAN topology[17] used in the simulation, which consists of three main parts: the distributed cell sites, the edge Cloud which is the one closer to the cell site, and the farther set of data centers. Any traffic routed from these sites towards a central entity will pass through a series of aggregation sites. The edges that connect different nodes in the DC backbone are weighted with propagation delays: one way latency from the cell site to a tier 1, tier 2 and tier 3 aggregation site are 0.6ms, 1.2ms and 4.2ms respectively [17]. The required computational resources of each network function are integers with a uniform distribution between 10 and 50, whereas the available resources of each substrate node varies between 100 and 150. We fix the price of utilization of DC servers for tier 1 \$50-40K, tier 2 \$30-25K and tier 3 \$20-10K ([18]). Substrate topology assumed to be connected with highly diverse links having capacity that ranges from 2000Gb/s down to 2 Gb/s and 1.25Gb/s for the wireless links.

uRLLC. For this slice, the placement algorithm should consider two important criteria. The first one is the latency and the second one is the cost; both should be minimized with a certain trade-off. We present a service request where the user plane traffic traversing the PHY, MAC, RLC and PDCP network functions has to be optimally placed in the substrate network (Figure 1) respecting the uRLLC constraints. By solving this instance we obtain a placement plan where the PHY, MAC, RLC and PDCP functions are placed at DC2 with minimum end-to-end latency of 0.6 ms between the endpoints where the cost of utilized data centers is equal to 196 units. By relaxing the latency constraints we obtain different split solutions with less cost as depicted in Figure 2 (a). This is achieved by converting the latency objective into a constraint, where the end-to-end latency must be less than a value specified by the user. In this scenario, we can see that the cost of the utilized DCs is 127 units respecting the latency constraint that must be less than 1.8ms and we obtain a placement plan with Split 2 (Figure 1) as a result. From these results, we may conclude that ensuring ultra-low latency has a cost, which in non negligeable for the network operators. The latter may reflect this cost when establishing the Service Level Agreement (SLA) with the Slice owner.

eMBB. Due to the huge packet size which requires high bandwidth in eMBB services, we tune our algorithm to maximize the data rate and minimize the cost of the utilized data centers. As a result, we obtain an optimal placement plan where PHY, MAC and RLC functions are located at DC2 and PDCP at DC6. According to the different objective implied in this use case, we obtain end-to-end latency of 1.8 ms. Network functions located in close proximity to users, reducing bandwidth requirements of backbone networks utilize same number of data centers but with cost of 91 units.

VI. Experimental Results

For the evaluation of the model we have performed the placement for different sets of deployment requests using the objectives defined. We have used the Gurobi Optimizer to solve the MIQCP and z3Opt to solve our MaxSAT instance on machines with Intel i7-6700 CPUs running at 3.40 GHz. As mentioned above our experimentation aims to evaluate two major factors (size of substrate network and split option)

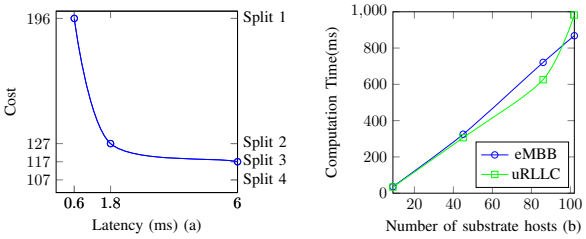

Fig. 2: (a) Functional split options for different latency (b) Impact of the substrate network size; requirements in uRLLC

affecting a total cost and computation time. Based on the specific service our evaluation results are divided into two major classes depicted by Figure 2 (b). Our placement results show that the total execution time of the MIQCP instance gradually increases with the increase in the number of substrate nodes and links.

For the evaluation of the MaxSAT model we have performed the placement for different sets of deployment requests using the objectives defined. Solutions obtained by the solvers are identical in case of uRLLC with MaxSAT and MIQCP formulations, even if the placement plans were different for some of the instances. This is explained by the equivalence of different placement plans.

Figure 3 is shows a simple RAN service request that includes PHY, MAC, RLC and PDCP. The network functions are modeled as radio function without configuration parameters. They act as a simple forwarding function. The results from our experiments show that the computational cost for providing formal assurance about reachability in addition to optimal embedding of virtual functions is greater than the case of MIQCP model. We conclude that the MaxSAT model is usually slower than the MIQCP model in finding optimal solutions, but it is more efficient in generating feasible solutions, especially for highly constrained and frequently encountered problems.

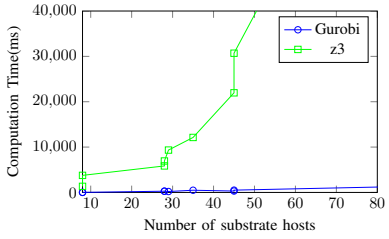

Fig. 3: Difference between MaxSAT and MIQCP approaches

VII. Conclusions and Future Work

In this paper we proposed an optimization framework for placement of RAN services based on a Mixed Integer Quadratically Constrained Programming (MIQCP) model and MaxSAT. First, we provided a comprehensive overview of implementation aspects and how different use case impact the implementation of RAN functional split. We further discussed different modeling approaches which may have a significant impact on complexity of the problem. Finally, we analyzed from a practical viewpoint, the actual flexibility that can be achieved in terms of functional split. Future work will be focusing on analyzing the different splits by looking at the actual models of radio functions and we plan to improve our abstract model to cope with bigger instances and use them to further scale our tool.

References

[1] N. Alliance, "5g end-to-end architecture framework," Tech. Rep., 2017, 04-Oct.

[2] G. T. 38.801, "Study on new radio access technology; radio access architecture and interfaces," Tech. Rep., 03 2017, v14.0.0.

[3] J. Y. Guido Marchetto, Riccardo Sisto and A. Ksentini, "Formally verified latency-aware vnf placement in industrial internet of things," in *14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy*, 2018, in press.

[4] U. Dtsch, 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, June 2013.

[5] 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.

[6] S. C. Forum, "Small cell virtualization: Functional splits and use cases," , January 2016, release 6.0.

[7] D. Harutyunyan and R. Riggio, "Flexible functional split in 5g networks," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov 2017, pp. 1–9.

[8] C. Y. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "Flexcran: A flexible functional split framework over ethernet fronthaul in cloud-ran," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[9] G. Mountaser, M. Condoluci, T. Mahmoodi, M. Dohler, and I. Mings, "Cloud-ran in support of URLLC," in *2017 IEEE Globecom Workshops, Singapore, December 4-8, 2017*, 2017, pp. 1–6.

[10] N. Makris, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas, "Experimental evaluation of functional splits for 5g cloud-rans," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[11] D. Sabella, P. Rost, A. Banchs, V. Savin, M. Consonni, M. D. Girolamo, M. Lalam, A. Maeder, and I. Berberana, "Benefits and challenges of cloud technologies for 5g architecture," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.

[12] O. Chabbouh, S. B. Rejeb, N. Agoulmine, and Z. Choukair, "Cloud ran architecture model based upon flexible ran functionalities split for 5g networks," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, March 2017, pp. 184–188.

[13] E. Pateromichelakis, K. Samdanis, Q. Wei, and P. Spapis, "Slice-tailored joint path selection scheduling in mm-wave small cell dense networks," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[14] P. C. Garces, X. C. Perez, K. Samdanis, and A. Banchs, "Rmsc: A cell slicing controller for virtualized multi-tenant mobile networks," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–6.

[15] O. Arouk, T. Turletti, N. Nikaein, and K. Obraczka, "Cost optimization of cloud-RAN planning and provisioning for 5G networks," in *ICC 2018, IEEE International Conference on Communications, 20-24 May 2018, Kansas City, MO, USA*, Kansas City, UNITED STATES, 05 2018.

[16] S. Spinoso, M. Virgilio, W. John, A. Manzalini, G. Marchetto, and R. Sisto, "Formal Verification of Virtual Network Function Graphs in an SP-DevOps Context," in *Service Oriented and Cloud Computing - 4th European Conference, ESOCC 2015, Taormina, Italy, September 15-17, 2015. Proceedings*, 2015, pp. 253–262.

[17] N. Alliance, "Overview on 5g ran functional decomposition," Tech. Rep. Version 1.0, 2018, 24-Feb.

[18] V. Suryaprakash, P. Rost, and G. Fettweis, "Are heterogeneous cloud-based radio access networks cost effective?" *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2239–2251, Oct 2015.